Kernel-Reliability-Based K-Means (KRKM) Clustering Algorithm and Image Processing

Chunsheng HUA†(a), Juntong QI(b), Jianda HAN†(c), Nonmembers, and Haiyuan WU†(d), Member

SUMMARY In this paper, we introduced a novel Kernel-Reliability-based K-Means (KRKM) clustering algorithm for categorizing an unknown dataset under noisy condition. Compared with the conventional clustering algorithms, the proposed KRKM algorithm will measure both the reliability and the similarity for classifying data into its neighbor clusters by the dynamic kernel functions, where the noisy data will be rejected by being given low reliability. The reliability for classifying data is measured by a dynamic kernel function whose window size will be determined by the triangular relationship from this data to its two nearest clusters. The similarity from a data item to its neighbor clusters is measured by another adaptive kernel function which takes into account not only the similarity from data to clusters but also that between its two nearest clusters. The main contribution of this work lies in introducing the dynamic kernel functions to evaluate both the reliability and similarity for clustering, which makes the proposed algorithm more efficient in dealing with very strong noisy data. Through various experiments, the efficiency and effectiveness of proposed algorithm have been confirmed.

key words: kernel reliability, adaptive kernel function, K-Means clustering

1. Introduction

For decades, clustering algorithms have been extensively studied as an elemental field of pattern recognition and played important roles in many computer vision tasks such as image segmentation, object tracking and recognition. Compared with the supervised algorithms, the clustering algorithms could classify an unknown dataset without the prior training process, which makes them suitable to work under the unknown condition. An effective clustering algorithm is expected to be able to classify the unknown data into groups by following their natural distribution, meanwhile rejecting the outliers, which is also the main purpose of this paper.

Among all the clustering algorithms, the conventional K-Means Clustering [31] (hereafter called as CKM) is the mostly well-known classifying algorithm that will unequivocally assign each vector of a dataset \( \{ x_1, x_2, ..., x_n \} \) into one of the manually selected \( K \) subsets (where \( K \) is usually a natural number). Besides its application in image segmentation [25], [28]–[30], the CKM algorithm has also been proved to be effective in object tracking [5], [26].

Further improvements on CKM algorithm were carried out by the Fuzzy C-Means (FCM) [11], [24] clustering algorithm which will generate a membership matrix \( M \) \( (M = \{ m_{11}, m_{12}, ..., m_{ik} \} \) \( i = 1 \sim m, k = 1 \sim K \) ). Each element \( m_{ik} \) of \( M \) indicates the membership from data \( x_i \) to cluster \( k \) \( (k = 1 \sim K) \). In FCM, a cluster center will be updated by computing the membership from all data to it. Hua [6] et al. improved the CKM by introducing the reliability evaluation into K-means clustering (hereafter called as RKM). Each data item will be assigned with a reliability value by computing the triangular relationship among the data and its two nearest neighbor clusters. In RKM, a true data will be assigned with high reliability and a noisy one will obtain low reliability. The success of RKM is based on the assumption that the noisy data are distant from all real clusters.

However, while applying the CKM or FCM into real applications, they usually suffer from two shortfalls: 1) the clustering result will be unreliable if the assumed number of clusters is incorrect; 2) all the data are assigned into one or more clusters without considering if they are the true data or not. Outliers are also processed as normal data, which leads the cluster centers to be attracted to outliers. And as for RKM, in the real applications, outliers are usually close to the true cluster centers, which results in the wrong output of RKM. Figure 1 shows an example that CKM, FCM and RKM all suffer from the strong noisy cluster which is very close to the true clusters. Their clustering results are updated towards the noisy cluster. The noisy cluster is defined as one cluster which is not assigned with initial cluster seed. In this paper, we call this as noisy cluster problem.

To reduce the affect of abnormal outliers while keeping the correct clustering results, in this paper, we brought out a novel Kernel-Reliability based K-Means (KRKM) clustering which will measure both the reliability and similarity for classifying an unknown data by the dynamic kernel functions. In KRKM, the outliers will be rejected for being assigned with low reliability which is derived from the dynamic kernel function measuring the triangular relationship among cluster seeds and data. The similarity between data and clusters is computed from an adaptive kernel function to ensure that each normal data could be categorized into the right cluster. We will explain the KRKM algorithm in detail.
in Sect. 3 and show its real application in Sect. 4.4

2. Related Work

To remove the affection of noisy data, Dave [9] brought out the noise cluster approach by introducing an additional noisy cluster to collect the outliers (an outlier is defined as a distant vector from all clusters). The distance between any vector and the noisy cluster is assumed to be the same and applied as the threshold. When the distance from a vector to all the clusters is longer than such threshold, it will be attracted to the noisy cluster and classified as outlier. The success of this method is based on the assumption that all the true clusters contain the same volume and the proper threshold is known as a prior. In the case of unknown dataset, it will be difficult for this algorithm to find the proper threshold. Jolion [20] brought out an approach called the Generalized Minimum Volume Ellipsoid Method (GMVE). In GMVE, after finding a minimum volume ellipsoid that covers (at least) h vectors of the data set X, the best cluster is determined by reducing the value of h gradually. This method suffers from the expensive computational cost and needs to specify several threshold parameters. Moreover, when the cluster shapes can not be described by the ellipsoids, this approach would not work.

The Possibilistic C Means Algorithm (PCM) [12], [13] computes a matrix of possibilistic membership P whose element \( p_{ij} \) will indicate the possibilistic membership of vector \( x_i \) belongs to the \( j^{th} \) cluster. The possibility is determined by the distance from \( x_i \) to the \( j^{th} \) cluster as well as the variance of \( j^{th} \) cluster. However, in the case that clusters are overlapping (or close to) each other, PCM tends to create low possibility to all data and get the wrong clustering results.

In [6], [7], the Reliability-based K-Means Clustering algorithm (RKM) was brought out to remove the noisy data. The reliability and possibility of classifying an unknown data into one group are measured by computing the triangular relationship among such data and its two nearest neighbor cluster centers. A true data will get high reliability, while the distant outliers will obtain low reliability and be rejected. The success of RKM is based on the assumption that outliers must be distant from any clusters. However, like (c) of Fig. 1, the result of RKM will be unreliable when the outliers are close to real clusters. Chintalapudi [17] proposed a Credibility Fuzzy C-Means method (CFCM). It assigns a credibility value for each vector \( x_i \) according the ratio of the distance between it and its nearest cluster to that between the farthest vector and its nearest cluster. However, if the farthest outlier is much farther than the rest outliers, this approach will assign high credibility values to most of the outliers.

Other variations of FCM have been reported [14]–[19], [21], [22] for reducing the affection of noisy data. Since covering all the related researches is not the main purpose of this paper, we refer good survey to [8] for more detailed discussion about the robust clustering algorithms under noisy condition.

3. Kernel Reliability Based K-Means Clustering

3.1 Kernel Reliability Estimation for Classification

As for a clustering algorithm, since the outliers will affect the update of cluster centers (in this paper we describe a cluster center as \( m_j \), \( j = 1 \sim K \)), it becomes important to check if an unknown data is an outlier or not before clustering it into any cluster. To reject the outliers either far or close to the real clusters, we introduce a dynamic kernel reliability estimation method to evaluate if it is reliable or not to process an unknown as the normal data.

As shown in Fig. 2, \( x_i \) represents an unknown data vector, \( m_{f(x_i)} \) and \( m_{n(x_i)} \) are the first and second nearest cluster centers to \( x_i \). The reliability for classifying \( x_i \) is determined through a dynamic kernel function by measuring the distance from it to \( m_{f(x_i)} \) and \( m_{n(x_i)} \) and that between \( m_{f(x_i)} \) and \( m_{n(x_i)} \) as following:

\[
R_{x_i} = \exp\left(-\frac{1}{r_{x_i}}\right) + \xi, \quad (1)
\]

\[
r_{x_i} = \frac{d_{f(x_i)}}{d_{j} + d_{i}}, \quad (2)
\]
considering the triangular relationship between it and its two nearest cluster centers (\(m_{f(x)}\), \(m_{s(x)}\)) through a dynamic kernel function.

Fig. 3 The reliability distribution of our KRKM and RKM [6] algorithms around one fixed cluster center. It shows that KRKM is more efficient in rejecting outliers close to real clusters. (Red curve: the reliability of RKM algorithm; Green curve: the reliability of our KRKM algorithm.)

For Fig. 3 and the following Fig. 4 are created under the same condition as: there are only two clusters and their cluster centers are fixed. Because when the cluster centers are fixed, it will be much easier to compute and describe the corresponding distribution of all data items around one fixed cluster center.

Fig. 4 The possibility distribution of the proposed KRKM and RKM algorithms around a fixed cluster center. In KRKM, only the data points close to cluster center will be assigned with high possibility. (Red curve: the RKM algorithm; Green curve: our KRKM algorithm.)

It means that the KRKM algorithm tend to only assign high reliability to the data which is close to the cluster center, while outliers will obtain low reliability as long as they are out of the real cluster.

3.2 Classification Possibility

The degree of a data vector \(x_i\) belonging to its first nearest cluster \(f(x_i)\) will be computed from a dynamic kernel function according to two distance \(d_f\), \(d_s\) as:

\[
\mu_{x_i} = \exp(-\frac{1}{\tau}y_i^2 + \xi),
\]

\[
\mu_{x_i} = \frac{d_s}{d_f + d_s},
\]

where, \(\eta = 3.0\). The possibility to classify \(x_i\) into its first nearest cluster \(f(x_i)\) can be computed as the product of \(R_{x_i}\) and \(\mu_{x_i}\) as:

\[
P_{x_i} = R_{x_i} \times \mu_{x_i},
\]

During the classification process, when \(P_{x_i} > 0.3\), data \(x_i\) will be assigned to its first nearest cluster, otherwise ignored. Figure 4 shows the possibility distribution (the \(P_{x_i}\) computed from Eq. (5)) of our KRKM and RKM algorithm around one cluster center. Compared with RKM (red curve), the proposed KRKM algorithm (green curve) will only assign high possibility to the data points close to the cluster center, which means it will be more insensitive to the outliers even if they are relative close to cluster center (such as the strong noisy data in Fig. 1).

3.3 Kernel-Reliability K-Means Clustering

The KRKM clustering will categorize the dataset \(X\) by minimizing the following objective function:

\[
J_{KRKM}(m) = \sum_{j=1}^{n} P_{x_i} \| x_i - m_{f(x_i)} \|^2.
\]

The cluster centers \(m\) could be obtained by solving the following equation:
Given the same dataset and initial clustering seeds as Fig. 1, KRKM algorithm successfully removes the affection of outliers and gets correct clustering result (red ⋄: clustering result; white ⋄: initial clustering seed, red →: the movement of cluster seed.).

\[
\frac{\partial J_{KRKM}(m)}{\partial m} = 0. \tag{7}
\]

When the Euclidean distance is assumed, the cluster center \( m_j \) will be firstly approximated as:

\[
m_j = \frac{\sum_{i=1}^{n} \delta_j(x_i)P_{x_i}x_i}{\sum_{i=1}^{n} \delta_j(x_i)P_{x_i}}, \tag{8}
\]

\[
\delta_j(x_i) = \begin{cases} 1 & \text{if } j = f(x_i) \\ 0 & \text{otherwise} \end{cases}. \tag{9}
\]

Then, we can get the \( m \) by using the results of Eq. (8) as the initial values with the Newton algorithm.

The KRKM clustering algorithm could be summarized as:

1) Initialization
   i) given the number of clusters \( K \)
   ii) given an initial value to each cluster center \( m_j, j = 1, \ldots, K \).

The initialization can be performed either manually or by using an external method (like K-means or Fuzzy K-means clustering).

2) Iteration of data grouping
   while \( m_j, j = 1, \ldots, K \) do not reach the fixed points, repeat the following steps.
   i) calculate \( f(x_i) \) and \( s(x_i) \) for each \( x_i \).
   ii) calculate the possibility for classifying \( x_i \) from Eqs. (1), (3), (5).
   iii) update \( m_j, j = 1, \ldots, K \) by solving Eq. (7).

Figure 5 shows the performance of KRKM on the same dataset as Fig. 1 where our KRKM algorithm successfully reduce the affection of outliers because they contain low reliability and possibility for classification according to Eqs. (1), (3), and (5).

### 4. Experiments and Discussion

#### 4.1 Convergence of the KRKM Algorithm

An important property of a clustering algorithm is whether it will converge or not. For such purpose, the objective functions of KRKM could be a good index to check its convergence. After a finite iteration number, if the value of objective function remains unchanged (or changes so small that such changes could be ignored), it means KRKM converges, otherwise diverges.

Here, we selected the IRIS\(^1\) and Aggregation datasets\([33]\) for evaluating the convergence of our algorithm. The IRIS dataset is a common dataset often used for testing the data grouping algorithm to check the convergence of our KRKM algorithm. The IRIS dataset contains 150 data which are divided into 3 groups and two of them are overlapping. Each group contains 50 data and each data has 4 attributes.

As for the Aggregation dataset, it contains 788 data distributed into seven clusters with different distribution. Each data contains two attributes.

Figure 6 shows the convergence evaluation of CKM, RKM and our KRKM algorithms under the same condition (the same test data, iteration and initial clustering data) with IRIS and Aggregation datasets. Since the objective functions of CKM, RKM and KRKM were different from each other, it is difficult for us to declare that KRKM converges better than other algorithms. However, as shown in the left and middle parts of Fig. 6, since the sum energy value of objective function becomes stable, it is clear that KRKM algorithm could converge as well as the rest two algorithms. The right image in Fig. 6 shows the distribution of Aggregation dataset and the corresponding clustering result of our KRKM algorithm. Since each data in IRIS dataset contains 4 attributes, we did not show it in the 2D graph.

#### 4.2 Affection of Parameters and Initialization

In this paper, we choose the value of parameter \( \tau \) and \( \eta \) experimentally so as to deal with the distribution real clusters and noise in our testing dataset. The affection of selecting different value of \( \tau \) and \( \eta \) has been shown in Fig. 7. Since increasing the value of \( \tau \) will lead to the rapid decrease of Eq. (1), only the data items that are close to the initial clustering seeds will gain relative high value and be remained, the others will be rejected for being regarded as the noisy ones. Therefore, the position of cluster centers are stopped since \( \tau = 6 \), because each cluster only contains one data left.

As shown in the bottom row of Fig. 7, as the value of \( \eta \) is increased, although the outcome of Eq. (3) will be decreased, the classification results will be little changed. That is because when the reliability value is correctly computed, the noisy data will be removed and all the reasonable data will be remained and treated equally. The possibility of classifying these data into their nearest clusters with Eq. (3) will be determined by their corresponding \( d_f \) and \( d_c \). In this way, the cluster center updated by Eq. (8) will change little even if \( \eta \) is increased.

Figure 8 shows the affection of initial positions of clus-

---

\( \text{http://www.ics.uci.edu/~mlearn/databases/iris/iris.data} \)
Fig. 6 The convergence of CKM, RKM and our KRKM algorithms with the “IRIS” and “Aggregation” datasets. The right image shows the clustering results of KRKM algorithm with the Aggregation dataset, where the white • represents the initial clustering seed and the red • indicates the clustering result.

Fig. 7 The affection of parameters in KRKM algorithm. Top row: when \( \eta \) is fixed, the clustering results as the value of \( \tau \) is increased; Bottom row: when \( \tau \) is fixed, the clustering results as the value of \( \eta \) is fixed. Red • indicates the clustering results; White • represents the initial clustering seeds.

Fig. 8 The affection of initialization in KRKM algorithm. It is clear that the performance of KRKM algorithm also depends on the positions of initial seeds. (a) (b): Good initial clustering seeds; (c) (d): Bad initial clustering seeds. Red • indicates the clustering results; White • represents the initial clustering seeds.

4.3 Performance Evaluation on the Simulated Dataset

Figure 9 shows the comparative experimental results among CKM, FCM, RKM and KRKM algorithms on the simulated dataset under different conditions. The white • means the initial clustering seeds and red • represent the final clustering results, the red → shows the movement of cluster seeds. Each row corresponds to different cases as: Row (1) – three initial clustering seeds are assigned to one cluster; Row (2) – two initial seeds are given to two separated clusters; Row
Fig. 9 Comparative experimental results among CKM, FCM, RKM and our KRKM algorithms. Column (a): results of K-Means Clustering; (b): results of RKM algorithm; (c): results of our KRKM algorithm; Red •: clustering results; White •: initial clustering seeds; Red →: the movement of cluster seed.

1. Row 1: three initial seeds are assigned to one clusters

2. Row 2: two initial seeds are assigned to two clusters

3. Row 3: two initial seeds are assigned to three clusters

4. Row 4: three initial seeds are assigned to five clusters

(3) – two initial clustering seeds are given to three separated clusters where two clusters are close to each other and one cluster is distant to them; Row (4) – three initial clustering seeds are given to five overlapping clusters.

In Row 1 of Fig. 9, the CKM algorithm forcibly divided one cluster into 3 parts because it treats all the data equally. FCM gets better results by evaluating the similarity between one cluster seed and all data without forcibly dividing them into groups. The RKM algorithm obtains the improved clustering results similar to the FCM by moving the cluster seeds toward one point. Since the kernel function will converge more locally, the KRKM got the best result that all clustering seeds are moved towards one position and they are almost completely overlapping each other.

As for Row 2 of Fig. 9, since the initial cluster seeds are separately placed at each real cluster, all the algorithms get almost the same results that they all successfully categorize the dataset into two separated clusters.

Row 3 of Fig. 9 shows a case that the number of initial cluster seeds is smaller than that of the real clusters in the dataset. Neither CKM nor FCM algorithm could get good results because they could not distinguish the noisy data from normal ones and their clustering results are attracted towards the noisy cluster. RKM achieves good clustering result by evaluating the triangular relationship among data and cluster seeds. Since the noisy data are distant from normal ones, they will obtain low reliability and be rejected by the RKM algorithm. The proposed KRKM algorithm gets the same good result as RKM by evaluating each data through the dynamic kernel functions.

Row 4 of Fig. 9 is a difficult case that three cluster seeds are assigned to five overlapping clusters where the cluster boundary is not clear. CKM algorithm divides the whole dataset into three parts by treating all data equally. While
Table 1  Performance evaluation of the CKM, FCM, RKM and KRKM of Fig. 9. Here, the KRKM won all the test, RKM ranks the second, while CKM and FCM achieve similar results.

<table>
<thead>
<tr>
<th>Noise rate</th>
<th>CKM (correction rate)</th>
<th>FCM (correction rate)</th>
<th>RKM (correction rate)</th>
<th>KRKM (correction rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 1</td>
<td>0%</td>
<td>33.3%</td>
<td>66.7%</td>
<td>66.7%</td>
</tr>
<tr>
<td>Row 2</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Row 3</td>
<td>33.3%</td>
<td>50%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Row 4</td>
<td>40%</td>
<td>0%</td>
<td>0%</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

FCM tends to consider the overlapping clusters as one huge cluster, therefore it moves all the cluster seeds towards one position. RKM could correctly move two cluster seeds to their corresponding cluster and wrongly update the third cluster seed at the middle of two clusters. That is because the overlapping among clusters is so serious that RKM algorithm considers the rest three clusters as one. While the KRKM algorithm is not affected by the overlapping problem because it will only compute the data near the kernel window, the noisy data out of the kernel window are ignored by dynamic kernel functions.

In Table 1, since all the clusters in Fig. 9 contain the same number of data, the performance of compared algorithms could be evaluated by the following criterion as:

\[
\text{noise rate} = \frac{\text{Num}_{\text{noisy clusters}}}{\text{Num}_{\text{all clusters}}}, \tag{10}
\]

\[
\text{correction rate} = \frac{\text{Num}_{\text{correct seeds}}}{\text{Num}_{\text{all seeds}}}, \tag{11}
\]

where, \(\text{Num}_{\text{noisy clusters}}\) means the number of noisy clusters in the dataset and \(\text{Num}_{\text{all clusters}}\) represents the number of all clusters, \(\text{Num}_{\text{correct seeds}}\) is the number of correctly classified cluster seeds, \(\text{Num}_{\text{all seeds}}\) indicates the number of all initial cluster seeds. After a finite iterations, when the distance between a cluster seed and a ground truth cluster center is smaller than a threshold (here, the variation value of that cluster), such cluster seed is considered as a correct seed, otherwise a wrong seed.

In order to evaluate the performance of KRKM with different data distribution, as shown in Fig. 10 and Fig. 11, we also applied it to the S-Dataset[32] and Spiral dataset[34]. The S-Dataset is composed of 4 datasets, where each dataset contains 5000 data items distributed in 15 clusters. The difficulty of these datasets is increased as the overlapping among clusters becomes more and more serious. The KRKM algorithm is successful in all clusters except one cluster in the S4 dataset (left bottom in (d)) due to its bad initial cluster seed that is located between two clusters. This also indicates that, just like CKM, the performance of KRKM also depends on the positions of initial seeds.

As for Fig. 11, the tested dataset is the Spiral dataset where the data items are distributed in the curve shape. Although the cluster seeds are updated towards the correct position by KRKM algorithm, when the classification result is displayed by different colors, it has been clear that the data items belonging to different clusters are wrong categorized into one group. This indicates that KRKM is unsuitable for the dataset whose data distribution does not follow the compact Gaussian distribution.

4.4 KRKM in Image Segmentation and Object Tracking

To test the performance of our KRKM algorithm in image segmentation and object tracking, we applied it into the framework of K-means tracker[5] which is also an object
tracking algorithm based on the dynamic image segmentation (details could be found in [5]).

Figure 12 shows the comparative image segmentation results of CKM [5], RKM [6] and the proposed KRKM algorithm. In (a) of Fig. 12, the red curves indicate the initialization of positive samples and green curves are the selected background (or negative) samples. Image pixels will be classified by different clustering algorithms between the positive and negative samples.

The (c) ~ (e) of Fig. 12 has shown the segmentation results of CKM [5], RKM [6], [7] and our KRKM according to the spatial and color similarity of each pixel. The total results of all compared algorithms are quite similar to each other, however there are still some differences and we illustrated them by the red circles. Here, since some background pixels contained similar color to the human hand, the CKM algorithm wrongly classified them into the target group. Although the RKM algorithm improved the segmentation quality by removing some distant noisy pixels, it still suffered from the strong noisy pixels that were both very close to the target and contained similar color to the skin. The performance of KRKM algorithm was much superior to the other two algorithms with the least segmentation error which was obtained by removing the strong noisy pixels with the kernel reliability and classification possibility estimation. That is because most segmentation error happens at the target boundary, which means the boundary parts can be considered as the mixtures of real data and noisy one. Since the KRKM algorithm can converge more locally, it can remove the noisy data from real ones.

In order to measure the segmentation accuracy of compared algorithms, the ground truth of tested image is manually created and the segmentation accuracy is measured as:

\[
\text{segmentation rate} = \frac{\text{Num}(\text{correct segment pixels})}{\text{Num}(\text{ground truth})},
\]

\[
\text{accuracy} = \frac{\text{Num}(\text{correct segment pixels})}{\text{Num}(\text{all segment pixels})},
\]

Table 2 Segmentation results of CKM, RKM and KRKM algorithms. All the methods contain similar segmentation rate, but the KRKM is superior to the other two algorithms in achieving the highest segmentation accuracy.

<table>
<thead>
<tr>
<th></th>
<th>CKM</th>
<th>RKM</th>
<th>KRKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation rate</td>
<td>89.6%</td>
<td>89.4%</td>
<td>89.1%</td>
</tr>
<tr>
<td>Segmentation accuracy</td>
<td>95.5%</td>
<td>95.9%</td>
<td>96.5%</td>
</tr>
</tbody>
</table>

The segmentation rate is a ratio between the number of correctly segmented pixels \((\text{Num}(\text{correct segment pixels}))\) and that of ground truth \((\text{Num}(\text{ground truth}))\). The segmentation accuracy is computed as the ration between the number of all segmented pixels \((\text{Num}(\text{all segment pixels}))\) and \((\text{Num}(\text{correct segment pixels}))\).

From Table 2, the segmentation rate of all methods is quite similar to each other, while the KRKM achieves the best accuracy as 96.5%.

Figure 13 shows the comparative experiments of object tracking among K-Means Tracker [5], RKM Tracker [6] and the proposed KRKM algorithm. Although all the trackers worked quite well on the top row, however, in the middle row, only a small part of the No.3 target was visible due to the occlusion, which could be considered as a case that a small true cluster is surrounded by the large strong noisy clusters. Neither the RKM nor K-Means tracker could track the No.3 target because they could not identify the correct cluster (here, it refers to the occluded person) under noisy condition. Therefore, it became clear that in frame 168, both the K-means and RKM tracker have had totally lost the No.3 target. K-Means tracker recovered from this failure in the last frame because the target person happened to move back to the position where it failed. Since the proposed KRKM tracker could remove the noisy data according to the kernel estimation, it is successful in tracking all the targets robustly through the test scene.

We also performed another comparative experiment with the PETS2006 dataset which is a well-known public...
dataset usually applied for object tracking and detection. As shown in Fig. 14, the difficulty of this test scene lies in that: among the four target persons, two of them are wearing the same color clothes and the clothes color of another person is also quite similar to them; meanwhile they also cross each other which makes it difficult to keep the correct correspondence to each target person. In the top row, the CKM tracker begins to lose one target person since frame 744 because the clothes color of that lost person is quite similar to his neighbor person. As for the RKM tracker, it becomes superior to CKM, but it lost the No.1 person since frame 835. That is because his neighbor person is wearing almost the same color cloth and the RKM tracker wrongly considers two persons as one group and update the cluster center to the wrong person. Only our KRKM tracker could track all the targets correctly through the whole scene, because even if the color of targets may be similar to his neighbor person, their position \((x, y)\) is still different. Therefore, such neighbor person could be considered as the noisy cluster in the 5D \((R, G, B, x, y)\) feature space (details about the 5D feature space could be found in [5]). Compared with the CKM and RKM tracker, our KRKM tracker is more powerful in removing the affection suffered from the noisy clusters, which guarantees the success of KRKM tracker.

All the experiments were taken on a desktop PC whose CPU is Intel Xeon 3.7Ghz and the memory is 32 GB. The tested image area for segmentation was about 400 × 400 pixels, and the processing time of KRKM algorithm was 747ms. As for the object tracking experiment where three targets exist and the image resolution is 960 × 540 pixels, the processing speed of proposed KRKM tracker is 19fps.

5. Conclusion

In this paper, we proposed a Kernel-Reliability based K-Means clustering algorithm for classifying a dataset under strong noisy condition. By estimating the classification reliability and possibility from the dynamic kernel functions according to the triangular relationship among a data and its two nearest clusters, strong/weak noisy data will obtain low values from those function and be rejected, while the true data can get high value and correctly classified. By applying the proposed algorithm to image segmentation and object tracking, the efficiency and effectiveness of proposed algorithm has been confirmed.

Acknowledgments

This work was supported by the National Key Technology R&D Program under Grant 2013BAK03B01.
Fig. 14 Comparative results of object tracking with PETS2006. Here, only the KRKM algorithm could track the targets correctly through all the test frames. The CKM [5] began to lose one target since frame 744 and the RKM [6], [7] trackers lost one target since frame 835. Top row: K-Means tracker [5]; Middle row: RK-Means tracker [6], [7]; Bottom row: the proposed KRKM tracker.

References

Chunsheng Hua received his B.E. degree in electronic engineering from Shenyang University of Technology in 2001. He received his MS degree from the Department of Mechanical and System Engineering at Kyoto Institute of Technology in 2004 and his Ph.D. degree in computer vision from the graduated school of system engineering at Wakayama University in 2007. From 2007 to 2010, he worked as postdoctoral researcher at the Institute of Scientific Industrial Research of Osaka University and was promoted to be a specially assigned assistant professor in 2010. Since October 2012, he joined the Shenyang Institute of Automation Chinese Academy of Sciences as a professor. He is a member of IPSJ. He received the Funai award of IPSJ Digital Courier for young researchers, Yamashita Memorial Award of IPSJ and SCIE award in 2006, 2008 and 2009, respectively. His research interests include machine learning, pattern recognition, clustering algorithms, object tracking, pedestrian detection and sensor fusion, etc.

Jianda Han received his Ph.D. from Harbin Institute of Technology in 1998. Currently he is a professor of the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include nonlinear estimation and control, robotics, and mechatronic systems.

Haiyuan Wu received the Ph.D. degree from Osaka University in 1997. From 1996–2002, she was the research associate at Kyoto Institute of Technology. She has been an associate professor and professor at Wakayama University in 2002 and 2008, respectively. She is a member of IPSJ, IEICE, ISCIE and the Human Interface.