International Journal of Artificial Intelligence and Applications for Smart Devices
Vol. 1, No. 1 (2013), pp. 15-24
http://dx.doi.org/10.14257/ijaiasd.2013.1.1.02

BH-centroids: A New Efficient Clustering Algorithm

Belal K. Elfarra[1], Tayseer J. El Khateeb[2] and Wesam M. Ashour[3]

[1]Dept. of Information Systems, Islamic University of Gaza, Palestine
[2]Dept. of Computer Center department, Islamic University of Gaza, Palestine
[3]Dept. of Computer Engineering, Islamic University of Gaza

[1]fbelal@iugaza.edu.ps

Abstract

The k-means algorithm is one of most widely used method for discovering clusters in data; however one of the main disadvantages to k-means is the fact that you must specify the number of clusters as an input to the algorithm. In this paper we present an improved algorithm for discovering clusters in data by first determining the number of clusters k, allocate the initial centroids, and then clustering data points by assign each data point to one centroid. We use the idea of Gravity, by assuming each data point in the cluster has a gravity that attract the other closest points, this leads each point to move toward the nearest higher gravity toward the nearest higher gravity point to have at the end one point for each cluster, which represent the centroid of that cluster. The measure of gravity of point (X) determined by its weight, which represent the number of points that use point X as the nearest point. Our algorithm employ a distance metric based (e.g., Euclidean) similarity measure in order to determine the nearest or the similar point for each point. We conduct an experimental study with real- world as well as synthetic data sets to demonstrate the effectiveness of our techniques.

Keywords: *Data clustering; G-means; PG-means; K-means algorithm; BH-centroids; Data mining*

1. Introduction

Clustering is a data mining (machine learning) technique used to place data elements into related groups. These groups can be of any shape and size that capture the most natural form of associated data. Several algorithms to classify data objects have been developed such as k-means clustering and expectation maximization (EM) clustering, in this paper we focus on the k-means algorithm as it is one of the most used iterative partitioned clustering algorithms and because it may also be used to initialize more expensive clustering algorithms (*e.g.*, the EM algorithm). However, k-means algorithm has two problem that it requires the user to specify the number of clusters (called k), and it suffers from initial starting conditions effects. It is not always clear what is the best value for k, and sometimes it will be difficult to choose k especially when the data have high dimension.

It is good to employ prior knowledge when choosing k, but sometimes prior knowledge may not exist this can make clustering less useful for exploratory data analysis in some case. In this paper we present a simple algorithm called BH-centroids (BH stands for black holes which has largest gravity) that discovers an appropriate k and provide initial centroids.

We describe examples and present experimental results that show that the new algorithm is successful. The rest of this paper is organized as follows. Section 2 briefly reviews the related

ISSN: 2288-6710 IJAIASD

work Section 3 describes the proposed algorithm in detail. Section 4 performs extensive experiments on the artificial and real dataset. Finally, conclusion and future work are presented in Section 5.

2. Related Work

Clustering algorithms can be classified into hierarchical and partitioning clustering algorithms [1]. Hierarchical algorithms decompose a database D of n objects into several levels of nested partitioning (clustering), represented by a dendogram, *i.e*. a tree that iteratively splits D into smaller subsets until each subset consists of only one object. In such a hierarchy, each node of the tree represents a cluster of D. Partitioning algorithms construct a flat (single level) partition of a database D of n objects into a set of k clusters such that the objects in a cluster are more similar to each other than to objects in different clusters. The Single-Link method is a commonly used hierarchical clustering method [2]. Starting with the clustering obtained by placing every object in a unique cluster, in every step the two closest clusters in the current clustering are merged until all points are in one cluster. Other algorithms which in principle produce the same hierarchical structure have also been suggested. Another approach to hierarchical clustering is based on the clustering properties of spatial index structures. The GRID [3] and the BANG clustering [4] apply the same basic algorithm to the data pages of different spatial index structures. A clustering is generated by a clever arrangement of the data pages with respect to their point density. This approach, however, is not well suited for high-dimensional data sets because it is based on the affectivity of these structures as spatial access methods. It is well-known that the performance i.e. the clustering properties of spatial index structures degenerate with increasing dimensionality of the data space. Recently, the hierarchical algorithm CURE has been proposed in [18]. This algorithm stops the creation of a cluster hierarchy if a level consists of k clusters where k is one of several input parameters. It utilizes multiple representative points to evaluate the distance between clusters, thereby adjusting well to arbitrary shaped clusters and avoiding the single-link effect. This results in a very good clustering quality. To improve the scalability, random sampling and partitioning (pre-clustering) are used. Optimization based partitioning algorithms typically represent clusters by a prototype. Objects are assigned to the cluster represented by the most similar (*i.e.*, closest) prototype. An iterative control strategy is used to optimize the whole clustering such that, *e.g.*, the average or squared distances of objects to its prototypes are minimized. Consequently, these clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if their number k can be reasonably estimated. Depending on the kind of prototypes, one can distinguish k-means, k-modes and k-medoid algorithms. For k-means algorithms, the prototype is the mean value of all objects belonging to a cluster. The k-modes [19] algorithm extends the k-means paradigm to categorical domains. For k-medoid algorithms, the prototype, called the medoid, is one of the objects located near the “center” of a cluster. The algorithm CLARANS introduced by [20] is an improved k-medoid type algorithm restricting the huge search space by using two additional user-supplied parameters. It is significantly more efficient than the well-known k-medoid algorithms PAM and CLARA presented in [21], nonetheless producing a result of nearly the same quality.

Determining k automatically has been extensively studied for many years. There is several algorithms have been proposed for this goal. Most of these algorithms wrap use splitting or merging rules to increase or decrease k until a proper value is reached. Pelleg and Moore [5] created X-means, an algorithm which determines the best k (out of a range of Ks) number of clusters for a dataset. This algorithm tries many values of k and uses Bayesian Information Criterion (BIC) to score each resulting model [11, 12]. The k that produces the highest BIC score is chosen. Besides BIC, other scoring systems, such as Akaike Information Criterion [6] and Minimum Description Length [14] can be applied. X-means is a straightforward extension of regular k-means. The difficulty it faces is: how many k values should be chosen and compared? When the data set is large and data distribution is non-trivial, the range of possible number of clusters can be large. G-means algorithm [15] is proposed to grow k from a small number. A statistical normality test is applied to each cluster to see whether it has high confidence of Gaussian distribution or not. If not, split the current cluster into two clusters and continue with the statistical test for the rest of the clusters. Like X-means, this algorithm is also a wrapper around k-means. It will generate a hierarchical tree of clusters. While the approach is intuitively meaningful, applying normality tests can become difficult when the set of data is extremely large (*e.g.* on the order of tens of thousands).

The one dimensional projection of the data will be very high in dimension and tend to look Gaussian according to the Central Limit Theorem and hence the need of splitting a cluster could not be detected even when it is not Gaussian. Powerful normality test like the Shapiro Wilk test [7] can handle a sample size of at most 5000. Also, the assumption of having Gaussian distribution in clusters is too strong in many real data, such as in Astronomy time series. It has been extensively tested within the LIGO community and it is known that LIGO data is not necessarily Gaussian in nature [8]. Sand and Moore [16] proposed an approach based on repairing faults in a Gaussian mixture model. Their approach modifies the learned model at the regions where the residual is large between the model's predicted density and the empirical density. Each modification adds or removes a cluster center. They use a hill-climbing algorithm to seek a model which maximizes a model fitness scoring function. However, calculating the empirical density and comparing it to the model density is difficult, especially in high dimension. Yu Feng Greg Hamerly [13] proposed PG-means which projects both the data set and learned clusters to one dimension and then applies the Kolmogorov-Smirnov test (KS) to check the goodness of fit of the data to distribution implied by the clusters where model parameters are learned by Expectation Maximization (EM). Tibshirani *et al.*, [4] proposed the Gap statistic, which compares the likelihood of a learned model with the distribution of the likelihood of models trained on data drawn from a null distribution. Our experience has shown that this method works well for finding a small number of clusters, but has difficulty as the true k increases. The primary contribution of this paper is a novel method of determining if a whole mixture model fits its data well, based on projections perform well in all situations; they tend to over fit, under fit, or are too computationally costly. These issues form the motivation for our new approach.

3. Methodology

Our algorithm is called BH-centroids, where BH stands for black holes. If we assume each data point creates gravity that stabilizes the relations between data points in the cluster, and by assuming the centroids as the black holes which have the largest gravity, then it will attract all nearest data point, and, as expected, each attracted point will pull its nearest data points to the centroid, at the end we have one data point which is the black hole. This point represents the centroid of the cluster. This idea can be implemented by determining the closest points X for each data point Y, and then compute the weights of these two data points W(X) and W(Y) by counting how many points use X as the closest data point, and so for Y. This weight is used to compute the data point gravity that the gravity increased as the weight of data point increased. After determining the weights of the two closest points, we let X to moves towards the nearest data point Y if, and only if, the following two conditions are satisfied: 1- the distance between

X and Y less than the threshold ε, 2- weight of X, W(X), is larger than W(Y). The threshold ε is the max distance between closest data points and used to prevent merging data points of different classes. By applying these steps we will have a good result, especially if the cluster has small width and height. But if the cluster shape has large width or height then it may have two BH centroids of distance greater than ε, for this, the algorithm will divide the cluster into two clusters. We can solve this problem by using vibration method to accumulate data points and to remove or minimize the effects of outliers. The vibration method is as follow: When X moves toward Y we apply equation (1) to all points of the dataset. We need to use vibration just with the first 2-3 iterations.

$$X(t+1) = X(t) + \big(Xj - X(t)\big)\eta \quad \text{where } (\eta <= 1) \tag{1}$$

Here we use η as a function of the distance between Xj and X(t), the value of η decreased -until it reach zero- as distance increased.

3.1. Clustering algorithm

BH-centroids algorithm is presented in Figure 1. It accepts as input the dataset Z, the algorithm begins by computing the k distance between all data points. And for each point x we define the closest point y, such that the distance between x and y not exceed the value ε. Then we move points to have same value by using the function 'Mov' described in figure 2. The function 'Mov' return the matrix with updated values. At the end of iterations we will have a matrix of points with values equal to one of expected centroids, at this point we can determine the number of clusters by counting the points of unique values, and we can determine the value of centroids by obtaining the unique values of points. For clustering, we use two methods: 1- By obtaining the number of clusters k and the value of centroids –as mention before- and then we use these values with the original dataset as input parameters to k-means algorithm for clustering data. 2- The other method is by comparing the result matrix x with the original dataset, that the result matrix x is already assign each point to the value of the centroid of the point's clusters. Table 2 shows the comparison between these two methods in clustering data.

```
Algorithm 1: BH-centroids (dataset Z)
$x = Z$;
$Compute\ k\ distances\ between\ all\ data\ points.$
$For\ i = 1 \rightarrow Length(X)$
    $if\ the\ closest\ point\ to\ X(i) is\ X(j)$   $\%with\ Euclidean\ distance\ [x(i), x(j)] <= \varepsilon$
        $x = mov(x(i), x(j))$;
    $end\ if$;
$end$
$repeat\ the\ algorithm\ until\ no\ change$
$return\ the\ matrix\ of\ the\ points\ with\ its\ new\ value.$
```

Figure 1. BH-Centroids Algorithm to determine the Number of Clusters

```
Function: Mov(a,b,x)
if $weight(a) > weight(b)$
    $b = a$;
    vibration $Dj$;
    for all $Db.distance < \varepsilon$
        $x = Mov(a, Dj)$;
    End loop
Else
    $a = b$;
    vibration $Di$;
    for all $Da.distance < \varepsilon$
        $x = Mov(b, Di)$;
    End loop
end if
return updated matrix $x$
```

Figure 2. Move Points Algorithm

Di is a subset contains all data points that use "x(i)" as the closest point. And Dj is a subset that contains all data points that use "x(j)" as the closest point.

4. Experiments

We perform several experiments on synthetic [10] and real-world [9] datasets to illustrate the utility of BH-centroids and compare it with k-means. For synthetic datasets, we experiment with Gaussian and non-Gaussian data. All our experiments use MATLAB version 7.6 on windows 7 on Intel core(TM) 2 Duo CPU T8300 2.40 GHz computer with 4 gigabytes of memory. The synthetic datasets used here in 2 dimensions with true ks are 4, 10, and 20 [10].

In the following Experiments we use BH-centroids to determine k and then we do comparison between k-means and BH-centroids in clustering data.

4.1. Synthetic Data Sets

Figure 3 shows the result of classifying data points by using k-means on 2-d synthetic dataset with 4 true clusters. The result was not very bad but with BH-centroids we have more accurate as shown in Figure 4.

Figure 5 shows the result of classifying data points by using k-means on 2-d synthetic dataset with 4 true clusters. BH-centroids correctly determines the k of clusters, however k-means classify some points of cluster B

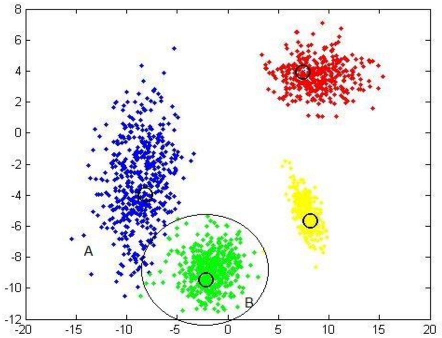

Figure 3. By using k-means as Classifier on 2-d Synthetic Dataset with 4 True Clusters. BH-centroids correctly chooses Four Centers

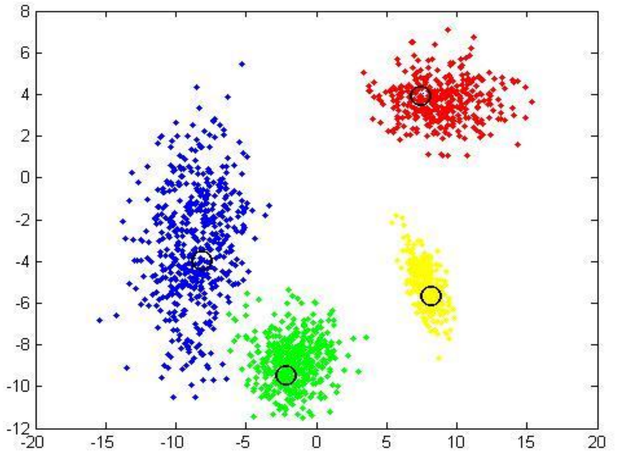

Figure 4. By using BH-centroids (ε = 3) as Classifier on Dataset used in Figure 1, the result is more accurate

as cluster A, in Figure 6 we have more accuracy by using BH-centroids as classifier.

Figure 7 shows the result of classifying data points by using k-means on 2-d synthetic dataset with 10 true clusters.

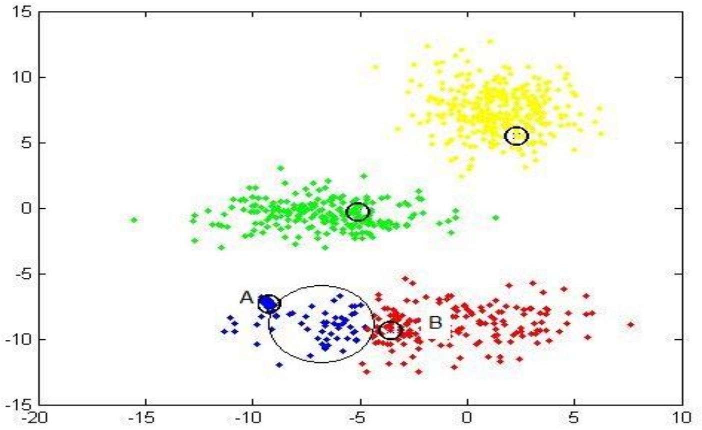

Figure 5. By using k-means as Classifier on 2-d Synthetic Dataset with 4 True Clusters, some points of Cluster B are assigned to Cluster A

BH-centroids determines k to be 11 that one of them is a noise (see Figure 7), data classified by k-means and as shown the classification is not good as in Figure 8 in which data are classified by BH-centroids.

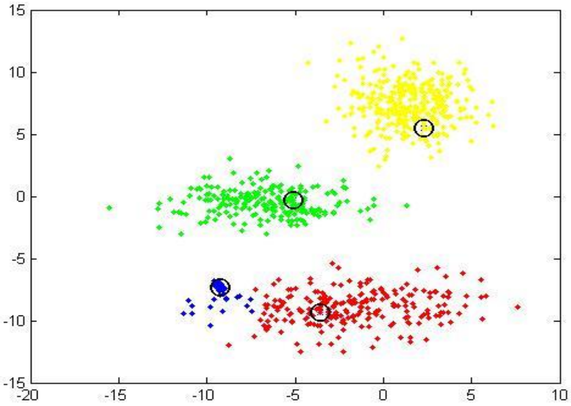

Figure 6. By using BH-centroids as Classifier on same Dataset used with Figure 3

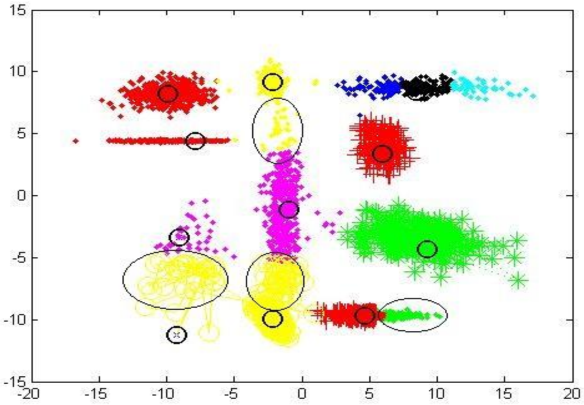

Figure 7. 2-d synthetic Dataset with 10 True Clusters.BH-centroids chooses 11 Centers

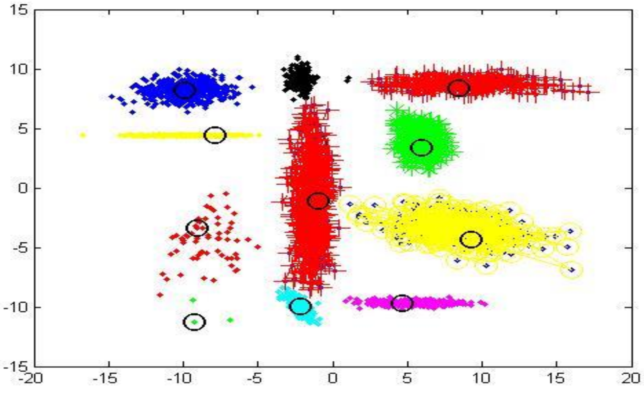

Figure 8. 2-d synthetic Dataset with 10 True Clusters. By BH-centroids Data Classified into 11 Clusters one of them has just Three Points

4.2. Real-world Data Sets

We experimented with three real-life datasets [9] whose characteristics are illustrated in Table 1.

Table 2. Real-life Datasets Characteristics

Dataset	No of Records	No of Attributes
Irise	150	4
Wine	178	13
lung-cancer	32	55

As in synthetic datasets, we apply BH-centroid first to determine k and the value of centroids then we apply the two methods mention before for clustering: k-means algorithm and our proposed method (BH-centroid), Table 2 shows the error rate of classification; we measure the error rate manual by comparing the result classification with the classification given by the dataset.

Table 2. The Error Rates of k-means and BH-centroids

Dataset	Determined clusters	K-means	BH-centroids
Irise	3	0.0067	0.0167
Wine	3	0.4261	0.3511
lung-cancer	3	0.2951	0.1875

5. Conclusions and Future Work

Each cluster can be represented by one centroid at which we have the largest weight with respect to other points in the cluster; in BH-centroids we consider this point as Black Hole which has largest gravity. All data points of one cluster will be attracted by the black hole; so every cluster will represented by one black hole. This is the idea of new algorithm BH-centroids for clustering data.

BH-centroids gives far more stable estimates of the number of clusters than existing methods over many different types of data of different shape and size.

However, we think this algorithm can be developed to give better result especially in determining the best value of the parameters η and ε.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure”, Proc. ACM SIGMOD’99 Int. Conf. on Management of Data, Philadelphia PA, **(1999)**.

[2] R. Sibson, “SLINK: an optimally efficient algorithm for the single-link cluster method”, The Comp. Journal, vol. 16, no. 1, **(1973)**, pp. 30-34.

[3] E. Schikuta, “Grid clustering: An efficient hierarchical clustering method for very large data sets”, Proc. 13th Int. Conf. on Pattern Recognition, vol. 2, **(1996)**, pp. 101-105.

[4] E. Schikuta and M. Erhart, “The bang-clustering system: Grid-based data analysis”, Proc. Sec. Int. Symp. IDA-97, vol. 1280 LNCS, London, UK, Springer-Verlag, **(1997)**.

[5] D. Pelleg and A. Moore, “X-means: Extending X-means with efficient estimation of the number of clusters”, Proceedings of the 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, **(2000)**, pp. 727-734.

[6] H. Akaike, “A new look at the statistical model identification”, IEEE Transactions on Automatic Control, vol. 19, no. 6, **(1974)**, pp. 716-723.

[7] S. Shapiro and M. Wilk, “An analysis of variance test for normality (complete samples)”, Biometrika, vol. 52, no. 3-4, **(1965)**, pp. 591-611.

[8] A. Abramovici, "LIGO: The laser interferometer gravitational wave observatory", Science, vol. 256, **(1992)**, pp. 325-333.

[9] UCI Machine Learning Repository < http://www.ics.uci.edu/ml/datasets.html>.

[10] Synthetic data for the evaluation of clustering algorithms<http://dbkgroup. org/handl/generators>.

[11] R. E. Kass and L. Wasserman, "A reference Bayesian test for nested hypotheses and its relationship to the schwarz criterion", Journal of the American Statistical Association, vol. 90, no. 431, **(1995)**, pp. 928-934.

[12] G. Schwarz, "Estimating the dimension of a model", The Annnals of Statistics, vol. 6, no. 2, **(1978)**, pp. 461-464.

[13] Y. Feng and G. Hamerly, "Learning the number of clusters in data", Department of Computer Science and Engineering Baylor University, Waco, Texas 76798.

[14] H. Bischof, Aleˇs Leonardis and A. Selb, "MDL principle for robust vector quantisation", Pattern analysis and applications, vol. 2, **(1999)**, pp. 59-72.

[15] G. Hamerly and C. Elkan, "Learning the k in k-means", Department of Computer Science and Engineering University of California, San Diego La Jolla, California 92093-0114.

[16] P. Sand and A. Moore, "Repairing faulty mixture models using density estimation", Proceedings of the 18th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA, **(2001)**.

[17] R. Tibshirani, G. Walther and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic", Journal of the Royal Statistical Society B, vol. 63, **(2001)**, pp. 411-423.

[18] S. Guha, R. Rastogi and K. Shim, "CURE: An Efficient Clustering Algorithms for Large Databases", Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, WA, **(1998)**, pp. 73-84.

[19] Z. Huang, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Tech. Report 97-07, UBC, Dept. of CS, **(1997)**.

[20] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. 20th Int. Conf. On Very Large Data Bases, Santiago, Chile, Morgan Kaufmann Publishers, San Francisco, CA, **(1994)**, pp. 144-155.

[21] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, **(1990)**.

Authors

Belal K. Elfarra received his B.Sc. degree in Computer Engineering in 2000 from Islamic University of Gaza (IUG) in Palestine. In 1999 he had worked as a programmer developer at PdMAIN Company in Gaza. He is currently working in Islamic university as a system analyst. He works in this paper towards his MS at the department of Computer Engineering at the Islamic University of Gaza. He focuses on multi-agent system, artificial intelligence and data clustering analysis.

Tayseer J. El Khateeb received his B.Sc. degree in Computer Engineering in 1999 from Islamic University of Gaza (IUG) in Palestine. Eng. Tayseer is currently work as a Head of computer Center department.

Wesam Ashour has received his B.Sc. degree in Electrical and Computer Engineering in 2000 from the Islamic University of Gaza. He has completed his M.Sc. in Multimedia with Distinction in 2004 from the University of Birmingham, UK. During his M.Sc. study, he has been awarded the prize for the best M.Sc. project 2003/2004. The project title is: Speech Recognition based on Lip Information. He has a PhD from the University of the West of Scotland in 2008. Dr. Ashour is currently a researcher at the Applied Computational Intelligence Research Unit in the University of the West of Scotland, UK since October, 2005. His research interests include data mining, artificial intelligence, reinforcement learning and neural networks.