TECHNICAL TRANSACTIONS 8/2018
ELECTRICAL ENGINEERING
DOI: 10.4467/2353737XCT.18.121.8896
SUBMISSION OF THE FINAL VERSION: 17/07/2018

Kazimierz Kiełkowicz orcid.org/0000-0001-5791-6069
kkielkowicz@pk.edu.pl
Department of Computer Science, Faculty of Electrical and Computer Engineering,
Cracow University of Technology

Unsupervised learning in latent space with a fuzzy logic guided modified BA

Uczenie nienadzorowane w przestrzeni utajonej z wykorzystaniem zmodyfikowanego algorytmu nietoperzowego sterowania rozmytego

Abstract

In this paper, a modified bat algorithm with fuzzy inference Mamdani-type system is applied to the problem of document clustering in a semantic features space induced by SVD decomposition. The algorithm learns the optimal clustering of the documents as well as the optimal number of clusters in a concept space; thus, making it suitable for a large and spare dataset which occur in information retrieval system. A centroid-based solution in multidimensional space is evaluated with a silhouette index. A TF-IDF method is used to represent documents in vector space. The presented algorithm is tested on the 20 Newsgroup dataset.
Keywords: document clustering, unsupervised machine learning, BA, fuzzy logic

Streszczenie

W publikacji zmodyfikowany algorytm nietoperzowy z rozmytym kontrolerem typu Mamdaniego został zastosowany do problemu analizy skupisk dla danych tekstowych. Proces uczenia odbywa się w przestrzeni skompresowanej, otrzymanej z dekompozycji SVD zbioru uczącego. Prezentowany algorytm uczy się jednocześnie optymalnego pokrycia klastrami przestrzeni oraz liczebności klastrów. Do oceny jakości rozwiązania zastosowano wskaźnik Sillhouette. Dane w reprezentacji wektorowej otrzymano z wykorzystaniem transformacji TF-IDF. Prezentowany algorytm przetestowana na zbiorze „20 Newsgroup".
Słowa kluczowe: klasteryzacja dokumentów, uczenie nienadzorowane, BA, logika rozmyta

1. Introduction

Unsupervised learning, also known as cluster analysis or classification, is a process of exploring the unknown structure of the data by separating a finite data set into clusters. Partitional unsupervised learning in particular divides the data object into a predefined number of clusters. Formally, in mathematical terms partitional unsupervised learning can be stated as follows [11]: given a set of m objects $X=\{x_1, \ldots, x_j, \ldots, x_m\}$, where $x_j=(x_{j1}, x_{j2}, \ldots, x_{jn}) \in R^n$, find a K-partition of X, $C=\{C_1, \ldots, C_K\}(K \leq m)$ such that $C_i \neq \emptyset$ for (i=1, …,K) and $\bigcup_{i=1}^{K} C_i = X$ and $C_i \cap C_j = \emptyset$ for $i,j=1,\ldots,K$ and $i \neq j$. The problem of unsupervised learning can be stated as the optimisation of a predefined criterion function so that data objects belonging to cluster C_i are more similar to each other than objects belonging to cluster C_j. In principal, such optimisation can be performed by a 'brute force' methods; however, in practice this is often unfeasible. An alternative approach is to use a heuristic algorithm like K-means. However, such a hill-climbing-like algorithm suffers from being sensitive to the initial starting point and is likely to get stuck in local minima. To overcome these limitations, it is advisable to use a more sophisticated metaheuristic algorithm.

In principal, metaheuristics algorithms can be divided into a few groups, e.g. algorithms based on an evolutionary approach that model evolutionary processes and algorithms exploring the phenomenon of swarm intelligence [6].

Another approach, such as algorithms for modelling the response of a human immune system (e.g. artificial immune system algorithms) might be considered as a separate category due to the multiplicity of these proposed solutions.

Metaheuristics methods which are focused on exploring models of natural evolution generally, although not exclusively, take the following forms: genetic algorithms (GA) [10], genetic programming (GP) and differential evolution (DE) [20, 21]. Algorithms based on swarm intelligence are broadly presented by particle swarm optimisation (PSO) [13], ant colony optimisation (ACO) [4] or some of its modifications.

In recent years, unsupervised learning based on natural inspired metaheuristic algorithms, including PSO [7, 14, 22] and ant algorithms [5] has attracted attention as a result of its demonstrated effectiveness in solving complicated optimisation problems.

The recently introduced method, based on the group of solutions which explore the phenomenon of swarm intelligence was presented by Yang [22] in 2010 and is called the bat algorithm (BA). In [22], by modelling the behaviour of bats hunting for prey and by exploring phenomena of their echolocation capabilities, the author managed to incorporate methods for balancing the exploration phase, as well as the exploitation phase of modern swarm based algorithms.

The BA have already been applied to solve numerous hard optimisation problems such as multi-criteria optimisation [23] or optimisation of the topology of microelectronic circuits [9]. The growing popularity of the BA has encouraged researchers to focus their work on its further improvement. Most work was done regarding the hybridisation of BA with other metaheuristics or local search methods [8]. Some other solutions were involved in the context of adding self-adaptability capabilities to algorithm [1]. Some work has also been done in the

area of the adaptation of the standard BA to binary problems [19] and in modifying scheme of acceptance of a new solution [16].

Unfortunately, most of these modifications not only improve the quality of the obtained solutions, but also increase the number of control parameters that need to be set to obtain solutions of an expected quality. This makes such solutions quite impractical. More recent work [15] introduces *a* fuzzy logic control system built on a Mamdani-type inference method to control the exploration and exploitation phases of an evolutionary system based on a modified BA [15]. The application of fuzzy logic to control the exploration and exploitation phases frees the user from explicit specifying control parameters and only requires the defining of expected behaviour in a knowledge-based form of *if-then* sentences that is readable for humans.

Section 2 describes a modified BA (MBA) used as an optimisation algorithm. Section 3 provides information about incorporating a fuzzy logic controller to dynamically adjust the behaviour of a MBA and define linguistic variables used in *if-then* sentences in the knowledge base. Section 4 describes used solution encoding as a Bat positing in the search space. Section 5 describes the used cost functions (Silhouette Coefficient index) that are used to evaluate a given solution. Section 6 provides information about inducing semantic feature space from raw data point space using singular value decomposition (SVD). Section 7 discusses methods to transform the raw text-based representation of documents to vector space used by the unsupervised learning algorithm. Section 8 provides experimental results conducted on subsets of the well-known 20 Newsgroup. Section 9 concludes this paper.

2. Modified Bat Algorithm (MBA)

The BA has been recently proposed as a bio-inspired metaheuristics method for solving hard real-valued optimisation tasks. It tries to mimic the behaviour of bats hunting for their prey. The algorithm was introduced by Yang in 2010 [22]. BA is based on a population of bats, which fly through and explore the solution search space in order to find interesting areas. Each single bat represents one solution in multi-dimensional search space. Solutions are evaluated in terms of their fit value by provided *a* fit function. The full discussion of BA, its shortcomings, and some proposed modifications to form a modified bat algorithm (MBA) can be found in [16]. A MBA is presented as pseudocode in algorithm 1:

1: Randomly initialise position x_i and velocity v_i of *i*-th bat in the population. A bat is an encoded solution as described in section 4.

2: Initialise pulsation frequency$Q_i \in [Q_{min}, Q_{max}]$, pulsation r_i and loudness A_i of *i*-th bat in the population.

3: **while not** termination conditions are satisfied:
Q = fuzzy inference system (diversity, error, iteration) Q adaptation using FLC

4: **for_each** bat in population:

5: $v_i(t) = \alpha_i v_i(t-1) + Q_i(x^* - x_i(t-1)) + Q_i(x^*_{ever} - x_i(t-1))$
$x_i(t) = x_i(t-1) + v_i(t)$

6: **if randn(0,1)** $> r_i^t$:
$x_i' \leftarrow$ generate new solution around current bat x_i

7: **if** $f(x_i') < f(x_i)$ **or randn(0,1)** $< A_i^t$:
$x_i \leftarrow x_i'$
Update values of pulsation and loudness, r_i^t and A_i^t, respectively, as:
$A_i^{t+1} \leftarrow \alpha A_i^t$; $r_i^{t+} \leftarrow r_i^t(1-\exp(-\gamma t))$

8: Evaluate bat population using the fit function *f* described in section 5.

9: Find the best bat in the population and mark it as x^*
10: **if** $f(x^*) < f(x^*_{ever})$:

11: $x^*_{ever} \leftarrow x^*$

Algorithm 1. Modification of bat algorithm to form MBA

where:
$v_i(t)$ – real-valued velocity vector of *i*-th bat,
$x_i(t)$ – real-valued position vector of *i*-th bat,
Q_i – pulsation frequency of *i*-th bat,
α, γ, Q_{min}, Q_{max} – constant.

The equations used for the bat position and the velocity update used in algorithm 1, step 5, were introduced in [22].

The modifications to BA introduced by Kiełkowicz and Grela in [16] are twofold: the scheme of acceptance of a new solution; a modified velocity equation to overcome some limitations of the original BA introduced in [22] and a memory of the best solution found during the process of optimisation by the algorithm is also introduced.

Modifications introduced in [16] also change the bat position and the velocity update equations. In comparison to equations presented in [22], the use of an archive component to help direct the bats towards the area where good solutions have previously been found and the concept of cognition coefficients instead of using upper bounds limits are used in [16]. Finally, equations (1) and (2) show the introduced modification:

$$v_i(t) = \alpha_i v_i(t-1) + Q_i(x^* - x_i(t-1)) + Q_i(x^*_{ever} - x_i(t-1)), \quad (1)$$

$$x_i(t) = x_i(t-1) + v_i(t). \quad (2)$$

where:
α_i – cognition coefficient of *i*-th bat,
$x^* - x_i(t-1)$ – social component,
$x^*_{ever} - x_i(t-1)$ – archive component,
Q_i – pulsation frequency of *i*-th bat.

In contrast to equations proposed by Yang in [22], modified velocity equation (1) uses cognition coefficients to limit the influence of past direction (taken at time $t - 1$) in the decision taken at current t iteration. There is also an archive component that helps bats build social knowledge of the previously found globally best solution.

The proposed modifications to the scheme of acceptance of new solutions tend to limit the probability of acceptance of the worse solution. Comparing the original BA with the modification in [16], the worse solution is accepted with probability A_i, where in the modified algorithm, the worse solution is accepted only with probability $(1 - r_i)A_i$. There is an obvious relation, that given $r_i > 0$ and $A_i > 0$, the following is true $(1 - r_i)A_i < A_i$. Moreover, modifications introduced in [16] also include the form of memory x^*_{ever} which represents the best solution ever found.

It is important that the introduced modifications do not change the computation complexity of the algorithm in the context of the big O notation since these modifications are linear in nature and are not based on additional computation or the evaluation of a fitness function.

3. Parameter auto-adaptation using fuzzy logic controller

The dynamic of the MBA is defined by the position and velocity update equations (1) and (2). Pulsation frequency Q_i was chosen to be dynamically adjusted using a fuzzy logic Mamdani-type inference system since this parameter has the most influence on the movement of bats in the colony. Dynamic changes of this parameter can improve the overall performance of the algorithm. However, it is not always possible to derive clear mathematical formula describing how parameters should be adopted during the optimisation process. It is easier to describe an expected behaviour of an algorithm in the form of an *if–then* human-readable sentence describing the situation and expected behaviour, e.g. "If the iteration is small, then exploration is intensive" or "if diversity is small and iteration is big, then exploration is less".

In the paper, a fuzzy logic Mamdani-type inference system is used to control the exploration/exploitation phase of a MBA through dynamic modification of the pulsation frequency Q_i, as it was introduced and analysed in the author's other publication [16].

To build a Mamdani-type inference system, it is required to define the input value range, the linguistic variable and the knowledge base in the form of an *if–then* sentence and define the output value range (and their defuzzification method). In the paper, as introduced in

[17], we also use diversity of the colony, the error of the flock and the number of iterations as the input parameters. As our output parameter, we choose Q_{max}. We expect all our input and output parameters to be in the range of

The diversity (dispersion) of the flock is defined by the following equation (3):

$$diversity(t) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{D}\left(x_{ij}(t) - x_j^*(t)\right)^2} \tag{3}$$

It can be considered as the average Euclidean distance between each bat and the bat representing the best solution at the i-th iteration. Diversity measures the degree of dispersion in the flock. When bats are close to each other, the diversity is small. Diversity needs to be normalised before it can be considered as input to a fuzzy inference system, since input must be in Equation (4) was used to normalise diversity:

$$normalizedDiversity(t) = \begin{cases} if\ \ minDiversity = maxDiversity,\ 0 \\ if\ \ minDiversity \neq maxDiversity,\ \dfrac{diversity(t) - minDiversity}{maxDiversity - minDiversity} \end{cases} \tag{4}$$

For iteration to be considered as input to *a* fuzzy logic inference system it needs to be normalised, we used formula (5):

$$Iteration = \frac{currentIteration}{\max imumNumberOfIteration} \tag{5}$$

From now on, *normalised Diversity(t)* will be referred to simply as *diversity(t)* and Q_{max} as *Q*. Membership functions defining input and output linguistic variables over crisp [0, 1] interval are the triangular functions shown in Figs. 1 and 2.

The knowledge base for Mamdani-type inference is in the form of a set of *if–then* sentences, where the *if* part is a premise and the *then* part is the conclusion. Each sentence is constructed using linguistic variables and (possibly) '*and/or*' connectors and hedges. In the paper, the author considers two linguistic variables (diversity and iteration) as inputs to the inference system and one output linguistic variable (Q). Each variable can take linguistic values from the set {small, big}. Input and output linguistic values are fuzzy sets defined on interval . Thus, we expect crisp input values and output to be in interval . The used knowledge base is as follow:

KnowledgeBase:

$$= \begin{cases} \text{"}\textbf{If}\ \text{diversity is } small \text{ and iteration is very } small, \textbf{then}\ \text{Q is very } big \\ \textbf{If}\ \text{diversity is } small, \textbf{then}\ \text{Q is } big \\ \textbf{If}\ \text{iteration is } big, \textbf{then}\ \text{Q is } small \\ \textbf{If}\ \text{iteration is very } big, \textbf{then}\ \text{Q is very } small \end{cases}$$

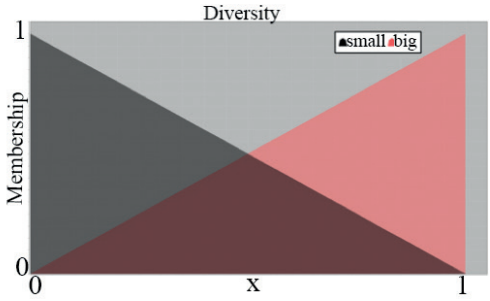

Fig. 1. Diversity linguistic variable and terms {small, big}

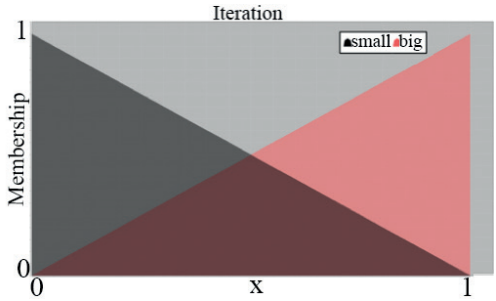

Fig. 2. Diversity linguistic variable and terms {small, big}

4. Solution encoding as bat position in dimensional space

One of the major design considerations in developing unsupervised learning algorithms based on the bat optimisation metaheuristic is encoding the solution as bat position in search space. In the context of clustering N data points into K clusters with cluster centroids $M=\{m_i\}$ $i=1,\ldots, K$ ($m_i \in R^n$ for learning in n-dimensional space), one way is for the bat position to directly encode the centroid positions as $bat_i:=(m_1, \ldots, m_K)$ and assign a data point to the cluster based on similarity measures, e.g. *the* Euclidean distance. Which is known in literature, for other types of population-based metaheuristics, as centroid-based representation [18]. However, in this paper the author augments bat representation with thresholds and encodes the *maximum* numbers of centroids (given as inputs for the algorithm) in the bat position. This way, the optimisation process searches for both the optimum number of centroids and the optimal centroids position in the search space regarding the given cluster index (in this paper, the silhouette coefficient is used). The solution encoded as the bat position in the search space is given as $bat_i:=(t_1,\ldots, t_K, m_1, \ldots, m_K)$, where threshold $t_i \in [-1,1]$, $m_i \in R^n$. If t_i satisfied $t_i > 0$ the corresponding centroid is considered active. All active centroids form a valid solution which is evaluated under the given clustering index (discussed in section 5).

5. Cost function

In literature, there are few cluster indices that can be used to evaluate clustering obtained by the optimisation algorithm, for example CH-index [2] or DB-index [3]. However, in this work, the silhouette coefficient [24] index is used as a cost function to evaluate a given solution found by the MBA. Since it evaluates how a given object (here a document) is similar to other objects (documents) in a cluster, it seems reasonable to use it in a document clustering problem where we are interested in having similar documents within a cluster.

In order to construct silhouettes, one needs two things: the partition obtained by the MBA and collections of all proximities between data objects. For each data object i within the data, we calculate the value $s(i)$. The mean value of all data objects is then calculated. Let us define $s(i)$ in terms of dissimilarities. Consider data object i in the data set and corresponding

cluster A to which data point *i* belongs. When cluster A contains other object thatwe define $a(i)$ as the average distance to all other data points in A.

Let us define $d(i,C)$ as the average dissimilarity from data point to all data points in the other cluster C ($A \neq C$). After obtaining all $d(i,C)$ for all other clusters in the solution, we can define $b(i)$ as:

$$b(i) = \min_{for\ every\ C \neq A} d(i,C) \tag{6}$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{7}$$

Using silhouette coefficients is useful when seeking compact and clearly separated clusters.

6. Latent semantic analysis using SVD

Before unsupervised learning can take place, the learning dataset must be transformed to a vector space of reduced dimensionality – the latent space. The transformation is linear and is based on singular value decomposition [12]. SVD factorises the given learning data (in the form of a matrix X_{mn} with m learning data points $x_i \in R^n$, $X_{mn} = [x_1; \ldots; x_m]$) to three matrices $X_{mn} = U_{mr} \Sigma_{rr} V^T_{rn}$, where U and V are orthogonal matrices $U^T U = V^T V = I$ and the diagonal matrix Σ contains r singular values of X_{mn}. The approximation of X is computed by setting all by the largest k singular values in Σ to zero ($=\widetilde{\Sigma}$), which is rank k optimal in the sense of L_2-matrix norm. Once the approximation $\widetilde{X} = U\widetilde{\Sigma}V^T \approx U\Sigma V^T = X$ is obtained, notice that the data-to-data inner products based in this approximation are given by $X\widetilde{X}^T = U\widetilde{\Sigma}^2 V^T$and thus the rows of $U\widetilde{\Sigma}$ are defining coordinates for data points in latent space. Once latent representation of a training data set is induced, unsupervised learning can take place.

7. Term frequency-inverse document frequency

In order to perform machine learning on text documents, we first need to convert text-based represented documents into numerical feature vector-space representation. There are numerous transformation methods known in the machine-learning community to perform such transformations of raw dataset text representation to vector representation. For example, 'bag of words' methods, which can be summarised as follows: first assign an integer number to each word occurring in any document in a training dataset, for example, by building a dictionary from words to integer indices. Then, for each document, count the number of occurrences of each word from the dictionary and store it in the matrices' representation as the number of appearances of each word from the dictionary in the document. However, dictionary representation has severe limitations. Consider the space requirements for storing matrices that were built using a dictionary with 100,000 different words on 10,000 documents in 32-bit float data type; this would require more than 4GB of RAM.

Transformation methods based on occurrence count have other severe limitations: longer documents have, by their nature, a larger average count than shorter documents, even though they might talk about exactly the same topics. To avoid these potential pitfalls, it suffices to divide the number of occurrences of each word in a document by the total number of words in the document: these new features are called term frequencies or tf for short.

Another refinement in addition to term frequencies is to weight the words that occur in many documents in the corpus and are therefore less informative than those that occur only in a smaller portion of the corpus. This weighting is called term frequency-inverse document frequency or tf–idf method for short. Words that are common in a single or a small group of documents tend to have higher tf–idf numbers than common words such as articles and prepositions. Search engines and modern information retrieval systems widely use tf–idf methods. Equation (8) represents the classical formula for tf–idf used for term weighting.

$$w_{i,j} = tf_{i,j} \log\left(\frac{N}{df_i}\right) \tag{8}$$

Where: $w_{i,j}$ is the weight for term in the documents j, N is the number of documents in collections, $tf_{i,j}$ is the term frequency of term i in document j and df_i is the document frequency of term i in the collections.

The tf–idf method evolved from IDF which was proposed by [25] and [26] with intuition that a term which occurs in many documents is not as good a discriminator as a term occurring in fewer documents and should have lower weight than the second.

8. Experimental results

The dataset used for simulation experiments is the 20 Newsgroup. This is a collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. It is a popular collation, widely used in text application experiments of machine learning techniques, such as text clustering. The data is organised into 20 different newsgroups, each corresponding to a different topic. Some of these topics are highly related, for example 'comp. sys.ibm.pc.hardware' and 'comp.sys.mac.hardware', while others are highly unrelated like 'talk.politics.mideast' and 'sci.crypt'.

For simulation experiments, a few datasets were constructed. The first dataset consists of documents from three different topics: 'soc.religion.christian', 'comp.graphics', 'rec. motorcycles' – this is referred to as DS1. The second dataset consists of dataset from four topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles' and 'rec.sport.hockey' and is referred to as DS2. The third dataset consists of five topics: 'soc.religion.christian', 'comp. graphics', 'rec.motorcycles', 'rec.sport.hockey' and 'comp.sys.ibm.pc.hardware' and is DS3. The fourth dataset consists of six topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles', 'rec.sport.hockey', 'comp.sys.ibm.pc.hardware' and 'sci.crypt' and is DS4.

After being loaded into memory, the datasets were transformed from text-based representation into vector-space representation of documents with the aforementioned tf-idf

method. Latent concept space was induced with SVD decomposition. The data was then normalised. On the normalised dataset, the modified BA was used to perform unsupervised machine learning. The algorithm not only returns the optimal partitioning of the given dataset, but also the optimum number of clusters given used clustering index silhouettes coefficient. The obtained solutions were compared with the kMeans algorithm for which the silhouettes coefficient was also calculated. Firstly, we performed experiments to see how the dimensionality of the concept space influenced the quality of the solutions; we then compare the proposed method with the kMeans algorithm.

During the simulation experiments, we used the knowledge base for dynamic parameter adjustments based on the current state of the colony for example, during the optimisation process using a Mamdani-type fuzzy inference system, the diversity and iteration number were as follows:

KnowledgeBase:

$$= \begin{cases} \text{"}\textbf{If}\ \text{diversity is } \textit{small} \text{ and iteration is very } \textit{small}, \textbf{then}\ Q \text{ is very } \textit{big} \\ \textbf{If}\ \text{diversity is } \textit{small}, \textbf{then}\ Q \text{ is } \textit{big} \\ \textbf{If}\ \text{iteration is } \textit{big}, \textbf{then}\ Q \text{ is } \textit{small} \\ \textbf{If}\ \text{iteration is very } \textit{big}, \textbf{then}\ Q \text{ is very } \textit{small} \end{cases}$$

For all experiments, the algorithm was set to stop after 5000 iteration steps; the number of bats was set to 1000**dimmensions*; and maximum number of cluster to discover was set to 6; α= 0.1, γ=0.85. Each test was re-run 50 times.

Figures 3 shows the mean solution found after 50 re-runs against a concept space dimensionality for a dataset containing three topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles' DS1.

Figures 4 shows mean solution found after 50 re-run against a concept space dimensionality for dataset containing four topics: "soc.religion.christian", "comp.graphics", "rec.motorcycles" and "rec.sport.hockey" DS2.

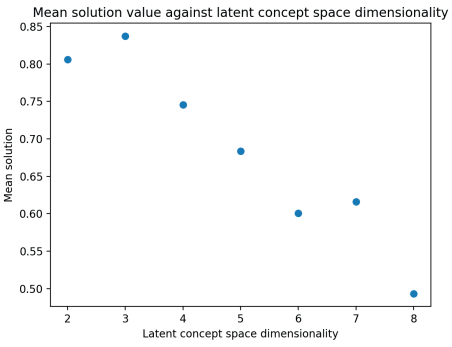

Fig. 3. Mean solution found for dataset DS1

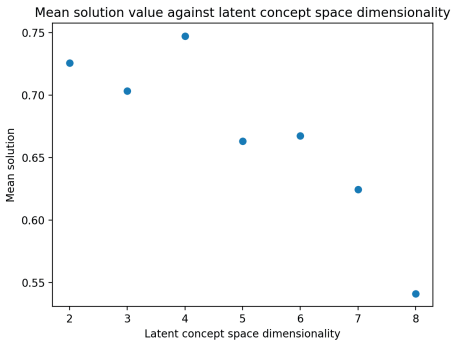

Fig. 4. Mean solution found for dataset DS2

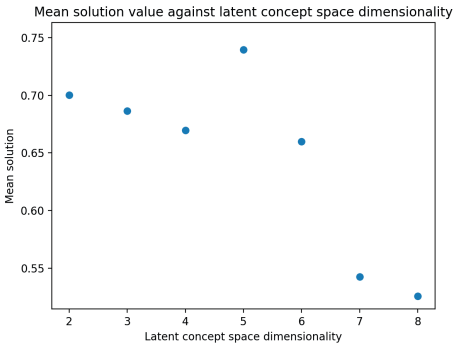

Fig. 5. Mean solution found for dataset DS3

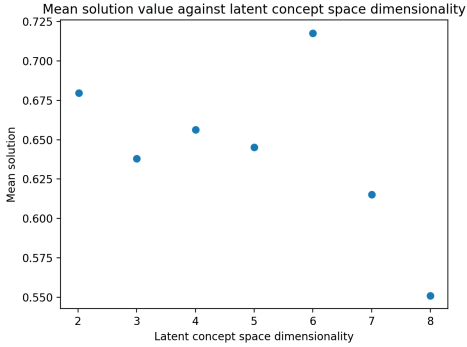

Fig. 6. Mean solution found for dataset DS4

Figure 5 shows the mean solution found after 50 re-runs against a concept space dimensionality for a dataset containing five topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles', 'rec.sport.hockey' and 'comp.sys.ibm.pc.hardware' DS3. Figure 6 shows the mean solution found after 50 re-runs against a concept space dimensionality for a dataset containing six topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles', 'rec.sport.hockey', 'comp.sys.ibm.pc.hardware' and 'sci.crypt' DS4.

The results presented in Table 1 show how the proposed algorithm performs on used test datasets and it is compared to the classic kMeans algorithm, for which the silhouette coefficient was calculated.

Table 1. Mean solution and standard deviation of the proposed algorithm compared to the kMeans algorithm

Dataset	Mean solution	Standard deviation	kMeans
DS1	0.849	0.096	0.755
DS2	0.751	0.145	0.714
DS3	0.745	0.163	0.656
DS4	0.715	0.193	0.516

9. Summary

The first modification to the BA are briefly discussed, then Mamdani-type inference system is shortly introduced and used linguistic variable and linguistic values have been defined. A full discussion of this modification and simulation experiments using different Mamdani-type inference systems for the dynamic modification of behaviour of a flock of bats in the BA can be found in the author's other publications [15] and [16]. This paper focused on the application of the proposed methods *in the document clustering problem of information retrieval systems.* An appropriate solution encoding the bat position in the search space is introduced which allows for both optimisation of cluster as well as their number.

Simulation experiments were conducted on the well-known 20 Newsgroup datasets. Since datasets, in their raw form, are word-based representations, they need to be transformed to vector-space representation before unsupervised machine learning can take place. The tf-idf transformation method was used for that transformation. Since the tf-idf method produces a sparse matrix, the concept space is induced using SVD decomposition.

A few tests were designed and conducted on different subsets of 20 Newsgroup original datasets. The first dataset (DS1) consists of documents from three different topics: 'soc. religion.christian', 'comp.graphics', 'rec.motorcycles'. The second dataset (DS2) consists of dataset from four topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles' and 'rec. sport.hockey'. The third dataset (DS3) consists of five topics: 'soc.religion.christian', 'comp. graphics', 'rec.motorcycles', 'rec.sport.hockey' and 'comp.sys.ibm.pc.hardware'. The fourth dataset (DS4) consists of six topics: 'soc.religion.christian', 'comp.graphics', 'rec.motorcycles', 'rec.sport.hockey', 'comp.sys.ibm.pc.hardware' and 'sci.crypt'.

The obtained results are reported, as well as their mean value and standard deviation for 50 re-runs of the proposed algorithm. The results obtained by the proposed method (MBA) were compared with the kMeans algorithms, which are well-known in literature. The mean solution found in terms of dimensionality of latent space is also considered.

The obtained results show that the proposed method is capable of finding a higher quality solution (in terms of the used clustering index silhouette coefficient) than the traditional method; therefore, it is better suited for the document clustering of information retrieval systems.

References

[1] Baziar A., Kavoosi-Fard A.A., Zare J., *A Novel Self Adoptive Modification Approach Based on Bat Algorithm for Optimal Management of Renewable MG*, Journal of Intelligent Learning System and Application, Vol. 5, Issue 1, 2013, 11–18.

[2] Caliński R., Harabasz J., *A dendrite method for cluster analysis*, Commun. Stat., Vol. 3, No. 1, 1–27, 197.

[3] Davies D., Bouldin D., *A cluster separation measure*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI–1, No. 2, 224–227.

[4] Dorigo M., Maziezzo V., Colorni A., *The ant system: optimization by a colony of cooperating ants*, IEEE Trans. on Systems, Man and Cybernetics B, Vol. 26, No. 1, 1996, 29–41.

[5] Dorigo M., Stützle T., *Ant Colony Optimization*, MIT Press, 2004

[6] Eberhart R. C., Shi Y., *Empirical Study of Particle Swarm Optimization*, 1999

[7] Eberhart R., Shi Y., *Particle swarm optimization: Developments, applications, and recourses*, Proc. Congr. EVol. Comput., 2001, 81–86.

[8] Fister I. Jr, Fister D., Yang X.S., *A Hybrid Bat Algorithm*, Elektrotehniski Vestnik, 2013, 1–7.

[9] Fong S., Yang X.S., Karamanglu M., *Bat Algorithm for Topology Optimization in Microelectronic Application*, International Conference on Future Generation Communication Technology (FGCT), IEEE, 2012, 150–155.

[10] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, 1989.
[11] Hansen P., Jaumard B., *Cluster analysis and mathematical programming*, Math. Program., Vol. 79, 1997, 191–215.
[12] Hofmann T., *Probabilistic Latent Semantic Analysis*, Proc of the Fifteenth conference on Uncertainty in artificial intelligence, UAI'99, 289–296.
[13] Kennedy J., Eberhart R.C., *Particle swarm optimization*, Proc. of IEEE International Conference on Neural Networks, Vol. 4, 1995, 1942–1948.
[14] Kennedy J., Eberhart R., Shi Y., *Swarm Intelligence*, Academic, 2001.
[15] Kiełkowicz K., Grela D., *FLC control for tuning exploration phase in bio–inspired metaheuristic*, Annales Universitatis Mariae Curie–Sklodowska, sectio AI – Informatica, 2017.
[16] Kiełkowicz K., Grela D., *Modified Bat Algorithm for Nonlinear Optimization*, International Journal of Computer Science and Network Security (IJCSNS), 2016, 46–50.
[17] Melin P., Olivas F., Castillo O., Valdez F., Soria J., Valdez M., *Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic*, Expert Systems with Applications, Vol. 40, Issue 8, 2013, 3196–3206.
[18] Merwe D., Engelbrecht A., *Data clustering using particle swarm optimization*, Proc. Congre. EVol. Comput, Vol. 1, 215–220.
[19] Mirjalili S., Mirjalili S. M., Yang Xin–She, *Binary Bat Algorithm*, Neural Computing and Applications, Vol. 25, Issue 3, 2014, 663–681.
[20] Storn R., Price K., *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*, Technical Report, 1995.
[21] Xu R., Xu J., Wunsch C., *A Comparison Study of Validity Indices on Swarm–Intelligence–Based Clustering*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS, Vol. 42, No. 4, 2012
[22] Yang X. S., A *New Metaheuristic Bat–Inspired Algorithm*, Nature Inspired Cooperative Strategies for Optimization, 2010, 65–74.
[23] Yang X. S., *Bat Algorithm for Multi-Objective Optimization*, International Journal of Bio-Inspired Computation, Vol. 3, Issue 5, 2011, 267–274
[24] Rousseeuw P., *Silhouettes: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Computational and Applied Mathematics 20, 53–65.
[25] Sparck Jones, K., *A statistical interpretation of term specificity and its application in retrieval*, Journal of Documentation, 28, 11–21.
[26] Sparck Jones, K., *IDF term weighting and IR research lessons*, Journal of Documentation, 60(6), 521–523.

If you want to quote this article, its proper bibliographic entry is as follow: Kiełkowicz K., *Unsupervised learning in latent space with a fuzzy logic guided modified ba*, Technical Transactions, Vol. 8/2018, pp. 141–154.