

Hybrid Binary Dragonfly Optimization Algorithm with Statistical Dependence for Feature Selection

Omar S. Qasim

Department of Mathematics,
University of Mosul, Mosul, Iraq.
E-mail: omar.saber@uomosul.edu.iq

Mohammed Sabah Mahmoud

Department of Mathematics,
University of Mosul, Mosul, Iraq.
Corresponding author: msmt_math@uomosul.edu.iq

Fatima Mahmood Hasan

Department of Mathematics,
University of Mosul, Mosul, Iraq.
E-mail: fatima.zamzoom@uomosul.edu.iq

(Received April 3, 2020; Accepted August 11, 2020)

Abstract

The aim of the feature selection technique is to obtain the most important information from a specific set of datasets. Further elaborations in the feature selection technique will positively affect the classification process, which can be applied in various areas such as machine learning, pattern recognition, and signal processing. In this study, a hybrid algorithm between the binary dragonfly algorithm (BDA) and the statistical dependence (SD) is presented, whereby the feature selection method in discrete space is modeled as a binary-based optimization algorithm, guiding BDA and using the accuracy of the k-nearest neighbors classifier on the dataset to verify it in the chosen fitness function. The experimental results demonstrated that the proposed algorithm, which we refer to as SD-BDA, outperforms other algorithms in terms of the accuracy of the results represented by the cost of the calculations and the accuracy of the classification.

Keywords- Feature selection; Classification; Dragonfly algorithm; Statistical dependence.

1. Introduction

Feature selection (FS) attempts to find the most discriminatory information aiming at designing an accurate learning system which depends heavily on the nature of the problem (Lu et al., 2017; Wang and Alexander, 2016). The selection of features can be formulated as a problem in combinatorial optimization since an exhaustive search can be impractical to find the optimal set of the features in a high-dimensional (HD) space. As a result, a subset of features that contributes to the best possible space separability is used to map the default dataset to a new one. The fitness function can be the accuracy of a given classifier or other criteria that can be considered the best trade-off between the computational cost of the extraction of the feature and the effectiveness of classification accuracy (Abualigah and Khader, 2017; Qasim and Algamal, 2020).

The dragonfly algorithm (DA) was developed by Mirjalili at Griffith University in (2016) (Rahman and Rashid, 2019). This technique, which is a swarm intelligence-based metaheuristic algorithm is inspired by the dragonfly's static and dynamic conduct in nature. There are two main optimization

stages: discovery and extraction dragonflies which were the basis of these two phases searching for food either dynamically or statically or avoiding the enemy. In dragonflies, there are two instances where swarm intelligence emerges from feeding and migration (Mirjalili, 2016). Feeding is in optimization is modelled as a static swarm; migration is modelled as a dynamic swarm. Because the DA algorithm can only solve continuous problems, it should be updated to solve binary problems which developed the basic DA algorithm by transforming the search algorithm in a discrete search space from the continuous search space. The modified DA algorithm operates on two-dimensional search areas (Chatra et al., 2019). The algorithm modified is called binary dragonfly algorithm (BDA), where particles are represented in binary space and their locations may take a binary 1 value or 0 value (Mafarja et al., 2018). There are several proposed swarm intelligence algorithms. PSO is an important swarm intelligence algorithm, and they have been used for nonlinear optimization (Pant et al., 2017). The GWO algorithm was used to calculate the optimization cost for the residual heat removal system. In addition, it is also used to calculate the optimal cost of the life support system in the space capsule. (Kumar et al., 2019a; Kumar et al., 2019b). An improvement of the chaotic crow search algorithm was proposed by Sayed et al. (2018). Due to the importance of the optimization algorithms in feature selection, Tawhid and Ibrahim presented in the year 2020 the whale optimization algorithm with other methods to obtain the best features and with a high classification accuracy (Tawhid and Ibrahim, 2020).

Statistical dependency (SD) is one of the filter methods calculated by using entropy among two discrete random variables. The entropy of a random variable calculates the mean information required to describe the random variable. This type of filter is used to arrange the features of datasets according to its importance in influencing the classification accuracy (Sugiyama and Borgwardt, 2013).

This study proposes a hybrid algorithm (SD-BDA) to determine the best features of genes. Hybrid proposed algorithm SD-BDA can focus on the power of both the SD and BDA algorithms by identifying and finding the most important genes in an effective manner (Alhafedh and Qasim, 2019; Dahiya et al., 2019).

The rest of this paper is organized as follows: The Feature Selection (FS) is presented in Section 2. The proposed algorithm SD-BDA is described in Section 3. Section 4 covers all the results obtained in this study. Finally, in Section 5, the most important general conclusions are elicited.

2. The Feature Selection (FS)

FS has different and multiple methods, some of which depend on finding the relationship between datasets characteristics such as the SD technique, and others depend on metaheuristic algorithm after their conversion from continuous space to discrete, such as the BDA, where the aim of all these techniques and algorithms is to reduce unnecessary features in dataset, which improves the accuracy of classification performance.

2.1 Statistical Dependence (SD)

The feature ranking method of statistical dependency (SD) is a more effective way of measuring the feature dependence with related class label values. First, each feature value dataset feature is determined into one of the quantization scale (QS) levels (Guilleminot and Soize, 2013; Song and Kang, 2009). The feature-specific quantization scale is calculated flexibly and effectively so that each bin can contain approximately equal samples throughout the entire dataset. In this method, the bins are selected instead of a traditional uniform QS to contribute to the occurrence of various

quantization levels by some statistical validity. The SD between the discrete value of the feature x and the class labels y is determined using Eq. (1) (Gretton et al., 2005).

$$D = \sum_{x \in X} \sum_{y \in Y} p(x, y) \frac{p(x, y)}{p(x)p(y)} \quad (1)$$

If Eq. (1) yields a larger value, the dependency between the values of the feature and the labels of the class is high. In case Eq. (1) produces a minimum value, this means that the features are fully independent of labels.

2.2 Dragonfly Algorithm (DA)

The DA algorithm, as the name indicates, was inspired by dragonflies, this algorithm is regarded as a technique of swarm intelligence (SI) to estimate the best solution (global) of a given problem of optimization (Hammouri et al., 2020; Mafarja et al., 2017). The dragonfly swarming behavior and the mathematical models are illustrated as follows:

- Separation refers to the process that is practiced by individuals to avoid collisions with other neighbors. Mathematically, this behavior is designed as in Eq. (2)

$$S_i = - \sum_{j=1}^N X_j - X_j \quad (2)$$

Where, x represents the current position, X_j is the neighboring j -th of the x position and N is the size of the neighbourhood.

- The alignment shows the velocity of the individuals is compared to other near-individuals. Mathematically, this behavior is designed as in Eq. (3)

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (3)$$

Where, V_j represents the velocity of the individual neighborhood, and N the size of the neighborhood.

- Cohesion refers to the tendency of the individuals to mass center of the neighborhood. As in Eq. (4), this behavior is modeled mathematically (KS and Murugan, 2017):

$$c_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (4)$$

Where, X represents the current position, X_j is the neighboring j -th of the X position and

N is the size of the neighborhood. “The two key behaviors each individual behaves to survive are attraction towards the food source and escape from enemies” (Mafarja et al., 2020) . The food attraction is modeled as in the Eq. (5).

$$F_i = X^+ - X \tag{5}$$

Where, X^+ is defined as the position of the food source and X is the position of the current individual:

$$E_i = X^- + X \tag{6}$$

Where, X^- represents the position of an enemy and X represents the position of the current individual.

DA used two basic vectors to solve optimization problems in the algorithm: the vector step and the vector position. The vector on the step is defined as:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w \Delta X_t \tag{7}$$

Where, s represents the weight of separation, the S_i shows the i – th individual separation, a is the weight of the alignment, A_i is the weight of the i – th individual c is the weight of the cohesion, C_i represents the cohesion of the i – th individual, f is food factor, F_i is food source of i – th individual, e represents the enemy factor, E_i is enemy position of the i – th individual, w is the weight of the inertia and t represents the number of iterations. The position of the DA is modified in a continuous search space by adding the current step vector to the previous position, however, the following equations should be used in a binary dragonfly algorithm (BDA) search space (Mafarja et al., 2017; Mafarja et al., 2018).

$$X_{t+1} = \begin{cases} -X_t & , r < T(\Delta X_{t+1}) \\ X_t & , r \geq T(\Delta X_{t+1}) \end{cases} \tag{8}$$

Where, r is a random in $[0, 1]$, $T(\Delta X_{t+1})$ is calculated as in Eq. (9).

$$T(\Delta X_t) = \left| \frac{\Delta X}{\sqrt{\Delta X^2 + 1}} \right| \tag{9}$$

The pseudocode of the BDA algorithm is displayed as follows:

Generate the initial population of DA , X_j & ΔX_j , $j = 1, \dots, N$

Generate an initial value A , a , and c .

Find the fitness function of each search agent.

while ($t < Max_{iter}$).

for each DA

Calculate the A , C & S by Eq. (2) to (4).

Update the E & F by Eq. (5) & (6) and the main coefficients.

Update the step vector by Eq. (7).

Calculate ΔX_{t+1} by Eq. (8).

3. Description of the Proposed Hybrid Algorithm SD-BDA

The hybrid method SD-BDA relies on the statistical dependence technique as an elementary stage to obtain a set of features, where the features of the data (from the highest importance to the lowest importance) are arranged according to their importance in the classification accuracy.

After arranging and specifying the features, they are entered into the BDA as a second stage, and a subset of pre-selected features with SD technique is selected. In BDA features are determined by relying on a binary value vector (consisting of one and zero) that is randomly generated with the same length as the features vector, and the feature that corresponds to the value of one is chosen, and which corresponds to zero is neglected, as shown in Figure 1:

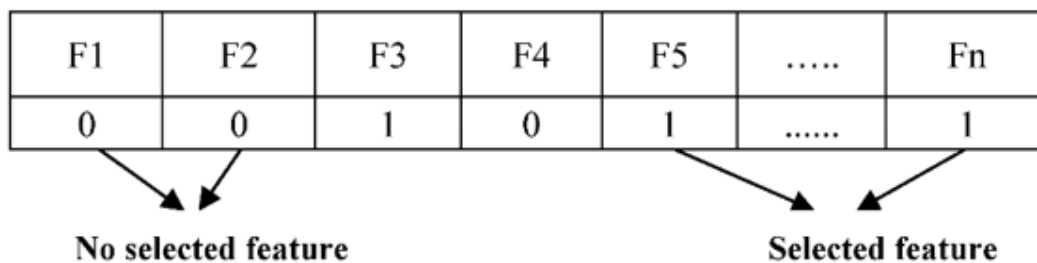


Figure 1. A representation of the features in BDA (Qasim and Algamal, 2020)

In the BDA, the KNN classifier is used to obtain classification accuracy, which is used in the fitness function, as it is defined as follows (Al-Thanoon et al., 2018; Alhafedh and Qasim, 2019).

$$Fitness\ Fun = \omega_1 * AC + (1 - \omega_1) * \left(\frac{F_q}{F_p - F_q} \right), \omega_1 \in [0, 1] \quad (10)$$

Where, AC is the accuracy of the classification, F_q is the selected feature, F_p is the all the features of dataset, and ω_1 is the random parameter corresponding to the weight of AC . The pseudocode of the proposed SD-BDA framework is displayed as follows:

Start

Generate the initial population of DA , X_j & ΔX_j , $j = 1, \dots, N$
 Generate an initial value A , a , and c
 Calculate the subset of features depending on SD .
 Find the fitness function of each search agent
while ($t < Max_{iter}$).
 for each DA
 Calculate the A , C & S by Eq. (2) to (4).
 Update the E & F by Eq. (5) & (6) and the main coefficients.
 Update the step vector by Eq. (7).
 Calculate ΔX_{t+1} by Eq. (8).
 Update the $T(\Delta X_t)$ Eq. (9).
 end for
 set $t=t+1$
end while
 Return the best position
 The optimal features.

End

4. Experimental Results and Discussion

In order to verify the hybrid algorithm, SD-BDA was applied to five different classification datasets (DLBCL, Lung, Prostate, Leukemia, and Ovarian), as all the datasets used are binary, obtained from the UCI repository (Bache and Lichman, 2013). The dataset was divided into 80% training groups, and a 30% test group of the total number of total data used in Table 1.

Table 1. Describe the types of datasets

Dataset	# Samples	# Features	Target class
Dataset1 (DLBCL)	77	7129	2
Dataset2 (Lung)	32	12533	2
Dataset3 (Prostate)	102	12600	2
Dataset4 (Leukemia)	76	7129	2
Dataset5 (Ovarian)	253	15154	2

The proposed algorithm SD-BDA was compared with both the BDA and the BGA in terms of the number of proposed features in addition to the classification accuracy, in addition, a 10-fold is set to obtain a reliable classification accuracy.

Table 2. A comparison between the SD-BDA and BDA in the average of feature selection

Dataset	# Feature selection SD-BDA	# Feature selection BDA	#All features
Dataset1 (DLBCL)	12.4	2950	7129
Dataset2 (Lung)	7.8	2589	12533
Dataset3 (Prostate)	5.6	3002.8	12600
Dataset4 (Leukemia)	12.6	3003.4	7129
Dataset5 (Ovarian)	5.8	6495.4	15100

Table 3. Experimental results, on average, of the SD-BDA, BDA, and BGA

Datasets	algorithms	Training dataset			Testing dataset
		CA	SE	SP	CA
Dataset1	SD-BDA	92.5459 (0.0270)	99.1571 (0.0115)	78.2662 (0.052)	88.8095 (0.10962)
	BDA	88.1159 (0.0121)	89.8235 (0.0091)	81.4945 (0.0342)	75.000 (0.000)
	BGA	85.5072 (0)	87.5 (0)	76.9231 (0)	75.000 (0)
Dataset2	SD-BDA	100 (0)	100 (0)	100 (0)	76.6667 (0.223)
	BDA	100 (0)	100 (0)	100 (0)	60.0000 (0.0912)
	BGA	100 (0)	100 (0)	100 (0)	63.3333 (0.217307)
Dataset3	SD-BDA	91.6893 (0.0099)	93.2194 (0.0292)	90.3328 (0.0131)	95.0476 (0.0353)
	BDA	65.4441 (0.015)	60.7795 (0.0102)	80.3684 (0.0262)	90.1905 (0.099)
	BGA	63.6216 (0.0129)	59.4979 (0.0076)	77.316 (0.0251)	50.9091 (0.0203)
Dataset4	SD-BDA	100 (0)	100 (0)	100 (0)	100 (0)
	BDA	100 (0)	100 (0)	100 (0)	77.8571 (0.1247)
	BGA	100 (0)	100 (0)	100 (0)	83.5714 (0.1506)
Dataset5	SD-BDA	97.6218 (0.0095)	97.9986 (0.0086)	96.9558 (0.0151)	96.1005 (0.0305)
	BDA	63.996 (0.0013)	63.9961 (0.0013)	66.64 (0.245)	64.1747 (0.0051)
	BGA	93.047 (0.0077)	96.6883 (0.0092)	87.3537 (0.0064)	62.3077 (0.0061)

It is clear from Tables 2 and 3, that the hybrid algorithm SD-BDA, gave better results through classification accuracy in addition to having chosen fewer features compared to the BDA, and this leads to a reduction in the cost of the calculations that the algorithm needs during the implementation process, where the training and testing dataset in the SD-BDA algorithm, achieved the preferable results for the classification accuracy. For instance, in Dataset 3, the accuracy of the testing dataset is 95.0476% by the SD-BDA which is higher than 90.1905% by BDA.

5. Conclusion

In this study, a hybrid algorithm was adopted that the features selection through two successive

phases. As the first stage relied on SD technique, it gave a subset of the features, whereas the second stage used a BDA to reduce the features formed from the first stage. The KNN classifier was adopted to evaluate the subsets of dataset in the fitness function. The results of the proposed hybrid algorithm SD-BDA were compared with both the BGA and BDA, as the SD-BDA demonstrated accuracy and superiority in classification performance by applying it to five different datasets.

Conflict of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The authors appreciate the effort from editors and reviewers. This paper did not receive any financial assistance/grants from any funding agencies. The authors sincerely appreciate the editor and reviewers for their time and valuable comments.

References

- Abualigah, L.M., & Khader, A.T. (2017). Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *The Journal of Supercomputing*, 73(11), 4773-4795.
- Al-Thanoon, N.A., Qasim, O.S., & Algamal, Z.Y. (2018). Tuning parameter estimation in SCAD-support vector machine using firefly algorithm with application in gene selection and cancer classification. *Computers in Biology and Medicine*, 103, 262-268.
- Alhafedh, M.A.A., & Qasim, O.S. (2019). Two-stage gene selection in microarray dataset using fuzzy mutual information and binary particle swarm optimization. *Indian Journal of Forensic Medicine & Toxicology*, 13(4), 1162-1171.
- Bache, K., & Lichman, M. (2013). UCI machine learning repository. University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>.
- Chatra, K., Kuppili, V., & Edla, D.R. (2019). Texture image classification using deep neural network and binary dragon fly optimization with a novel fitness function. *Wireless Personal Communications*, 108(3), 1513-1528.
- Dahiya, B.P., Rania, S., & Singh, B.P. (2019). A hybrid artificial grasshopper optimization (HAGOA) meta-heuristic approach: a hybrid optimizer for discover the global optimum in given search space. *International Journal of Mathematical, Engineering and Management Sciences*, 4(2), 471-488.
- Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005, October). Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory* (pp. 63-77). Springer, Berlin, Heidelberg.
- Guilleminot, J., & Soize, C. (2013). On the statistical dependence for the components of random elasticity tensors exhibiting material symmetry properties. *Journal of Elasticity*, 111(2), 109-130.
- Hammouri, A.I., Mafarja, M., Al-Betar, M.A., Awadallah, M.A., & Abu-Doush, I. (2020). An improved dragonfly algorithm for feature selection. *Knowledge-Based Systems*, 203, 106131.
- KS, S.R., & Murugan, S. (2017). Memory based hybrid dragonfly algorithm for numerical optimization problems. *Expert Systems with Applications*, 83, 63-78.

- Kumar, A., Pant, S., & Ram, M. (2019a). Gray wolf optimizer approach to the reliability-cost optimization of residual heat removal system of a nuclear power plant safety system. *Quality and Reliability Engineering International*, 35(7), 2228-2239.
- Kumar, A., Pant, S., Ram, M., & Chaube, S. (2019b). Multi-objective grey wolf optimizer approach to the reliability-cost optimization of life support system in space capsule. *International Journal of System Assurance Engineering and Management*, 10(2), 276-284.
- Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, 256, 56-62.
- Mafarja, M., Aljarah, I., Heidari, A.A., Faris, H., Fournier-Viger, P., Li, X., & Mirjalili, S. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Systems*, 161, 185-204.
- Mafarja, M.M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. Paper presented at the 2017 *International Conference on New Trends in Computing Sciences (ICTCS)* (pp. 12-17). IEEE. Amman, Jordan.
- Mafarja, M., Heidari, A.A., Faris, H., Mirjalili, S., & Aljarah, I. (2020). Dragonfly algorithm: theory, literature review, and application in feature selection *Nature-Inspired Optimizers* (pp. 47-67): Springer, Cham.
- Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053-1073.
- Pant, S., Kumar, A., Bhan, S., & Ram, M. (2017). A modified particle swarm optimization algorithm for nonlinear optimization. *Nonlinear Studies*, 24(1), 127-138.
- Qasim, O.S., & Algamal, Z.Y. (2020). Feature selection using different transfer functions for binary bat algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, 5(4), 697-706. doi:10.33889/ijmems.2020.5.4.056.
- Rahman, C.M., & Rashid, T.A. (2019). Dragonfly algorithm and its applications in applied science survey. *Computational Intelligence and Neuroscience*, 2019, 21. doi:10.1155/2019/9293617.
- Sayed, G.I., Darwish, A., & Hassanien, A.E. (2018). A new chaotic whale optimization algorithm for features selection. *Journal of Classification*, 35(2), 300-344.
- Song, J., & Kang, W.H. (2009). System reliability and sensitivity under statistical dependence by matrix-based system reliability method. *Structural Safety*, 31(2), 148-156.
- Sugiyama, M., & Borgwardt, K.M. (2013). Measuring statistical dependence via the mutual information dimension. Paper presented at the *Twenty-Third International Joint Conference on Artificial Intelligence*, China.
- Tawhid, M.A., & Ibrahim, A.M. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. *International Journal of Machine Learning and Cybernetics*, 11(3), 573-602.
- Wang, L., & Alexander, C.A. (2016). Machine learning in big data. *International Journal of Mathematical, Engineering and Management Sciences*, 1(2), 52-61.

