Intelligent Water Drops Algorithm for Rough Set Feature Selection

Basem O. Alijla[1], Lim Chee Peng[2], Ahamad Tajudin Khader[1], and Mohammed Azmi Al-Betar[1,3]

[1] School of Computer Sciences, Universiti Sains Malaysia, Pinang, Malaysia
[2] Centre for Intelligent Systems Research, Deakin University, Australia
[3] Department of Computer Science, Jadara University, Irbid, Jordan

Abstract. In this article; Intelligent Water Drops (IWD) algorithm is adapted for feature selection with Rough Set (RS). Specifically, IWD is used to search for a subset of features based on RS dependency as an evaluation function. The resulting system, called IWDRSFS (Intelligent Water Drops for Rough Set Feature Selection), is evaluated with six benchmark data sets. The performance of IWDRSFS are analysed and compared with those from other methods in the literature. The outcomes indicate that IWDRSFS is able to provide competitive and comparable results. In summary, this study shows that IWD is a useful method for undertaking feature selection problems with RS.

Keywords: Feature Selection (FS), Rough Set (RS), Intelligent Water Drops (IWD)

1 Introduction

Feature Selection (FS) refers to the process of selecting the minimum subset of features that preserves the meaning of the original features [3]. An irrelevant feature is a feature that is weakly correlated to the decisional feature, which can be removed with little or no effect to the given outcomes. A redundant feature is a feature that is highly correlated with other features, and it does not carry significant knowledge when it is added to the entire set of features. If the irrelevant and redundant features can be removed, the dimension of the data set can be reduced without significantly affecting the knowledge represented by the entire features [19]. Moreover, learning and classification accuracy can be improved by simple, easy, and understandable presentation of the underlying rules, which are formulated from fewer numbers of features [10].

The main elements of an FS algorithm include subset generation, subset evaluation, and the stopping criterion [9]. Subset generation is the search technique, which is used to explore the search space. Subset evaluation then uses an evaluation approach to assess the goodness of the subset of features. The stopping criterion is used to terminate the search process. FS problem is a combinatorial NP-hard problem [19]. This is because the numbers of alternatives are proportional to the number of features in the data set. As an example, if we have a data set with N features, FS can be seen as a search process over a search space with 2^N possible subsets of features. Although exhaustive search techniques can be used to find the optimal subset of features, it is impractical in the presence of large number of features. To manage the complexity of the search process, many search strategies such as heuristic and metaheuristic methods have been proposed [1, 5, 8, 17, 18]. A detailed taxonomy and the associated algorithms of FS can be found in [9].

Rough Set Theory (RST) is a mathematical theory introduced by Pawlak in 1982 [12], which was used as a tool for analyzing incomplete or uncertain data. RST is popular for feature selection. It is characterized by its ability to evaluate features indiscernibility without needing any external information. Indeed, RST is used to analyze only the hidden information within a data set to find the minimal knowledge representation. RST has been successfully used with many search algorithms for feature selection in order to measure the goodness of the selected subsets [1, 5, 8, 17, 18].

The Intelligent Water Drops (IWD) algorithm is a meta-heuristic method introduced by Shah-Husseini [16]. It is a nature-inspired optimization algorithm. IWD imitates some of the natural phenomena of a swarm of water drops with the soil onto the river bed. Within the last 5 years, IWD has been very successful in many discrete optimization problems [4, 6, 11, 13, 16] and machine learning tasks [15]. IWD has recently been adapted for continuous optimization problems [14]. This success is partly owing to the fundamental advantages of IWD over other traditional optimization techniques [13, 15, 16]. IWD has a simple and easily understandable mathematical model. It can be adapted easily for many optimization problems, and is applicable to both discrete and continuous problems. It converges fast to the optimal solution. It considers the construction of solution in the population based on information given by experience (gained from the previous iteration of the search) rather than considering refinement of the existing population.

In this paper, the IWD is adapted with the rough set for feature selection. The resulting model is called Intelligent Water Drops algorithm for Rough Set Feature Selection (IWDRSFS). IWDRSFS is evaluated with 6 benchmark data sets obtained from [7]. Many of these data sets come from the UCI machine learning repository [2]. The numbers of the input features vary between 13 and 56. The results from IWDRSFS are compared with those from local search-based methods, such as hill climbing, as well as population search-based methods such as ant colony and the genetic algorithm.

The rest of the paper is organized as follows. Section 2 provides a brief introduction to RST and RS dependency for feature selection. Section 3 describes the detailed modeling and implementation of IWD for feature selection. The experiments and the associated results are presented in section 4. Conclusions and suggestions for future work are highlighted in section 5.

2 Rough set theory and feature dependency for feature selection

RST is an approximation approach developed to deal with incomplete knowledge Pawlak [12]. The fundamental concept of RST is the approximation of the uncertain set (knowledge) with a pair of precise sets, called the lower and upper approximations. The lower approximation is a set that describes objects that are definitely belonging to the subset of interest, while the upper approximation is a set that describes objects that are possibly belonging to the subset of interest. The pair of lower and upper approximations as a tuple is defined as a Rough Set (RS)[12].

Let $IS = (U, A \bigcup D)$ be an information system, U is a non-empty finite set of objects (universe), A is a non empty finite set of conditional features, and D is the decisional feature. For any $S \subseteq A$ there exists an equivalence relation called the S-indiscernibility relation that can be used to group objects into classes which are called equivalence classes denoted as $[x]_s$. Each class contains the set of objects that have the same vector of features values in S. Let $X \subseteq U$. X be a target equivalence class (concept) induced by the decisional feature D. X cannot be expressed directly by $[x]_S$ because X may include an object that is not in $[x]_S$ and vice versa. RST is able to approximate this uncertainty by comparing the equivalent classes induced by the conditional features with the target equivalence class. RST defined the *lower* and *upper* approximations to find the positive region, which is a set that includes objects that can certainly be classified by a feature or subset of features.The positive region can be employed to find feature dependency, and is denoted as $\gamma_S(Q)$. $\gamma_S(Q)$ is used to measure the strength of the relation (correlation) between two set of features S, and Q. If $\gamma_S(Q) = 1$, then Q is totally dependent on S, and denoted as $(S \Rightarrow Q)$. If $\gamma_S(Q) < 1$ then Q is partially dependent on S with a degree $\gamma_S(Q)$, and is denoted as $(S \underset{\gamma_S(Q)}{\Longrightarrow} Q)$. Finally, if $\gamma_S(Q) = 0$, then Q and S are independent. The detailed information on RS can found in [9].

The main idea of FS with RS is to remove features that do not have significant effects on feature dependency. So, the FS algorithm aims to search for the minimum subset of features, S, that has feature dependency equals to the dependency of the full features C, i.e. $\gamma_S(Q) \approx \gamma_C(Q)$, where $S \subseteq C$.

FS problems require finding one subset of features that has feature dependency equals to the dependency of the full set of features. The ideal FS algorithms aim to find all subsets of features that satisfy the abovementioned condition. However, finding all subsets of features is computationally expensive. Therefore, an efficient search algorithm is required to find the optimal subset of features by considering the maximum dependency and minimum subset size.

3 Intelligent water drops for feature selection

In nature, water drops have to overcome obstacles and barriers in the environment in order to find the shortest path from its source to the destination. Water drops prefer to follow the direction of the easy path, i.e. a path with less soil. Water drops are transferred from one point to another with a velocity. During the move, water drops carry an amount of soil gained from the bed of the path. Changes on the soil carried by the water drops, the soil in the path, and the velocity, encourage water drops to move through the shortest path that has less soil and, at the same time, to reinforce other water drops to follow the same path.

The key properties of water drops are soil and velocity. During the trip of the water drops, a certain amount of soil from the bed of the path will be carried together. The change on the soil carried by the drops is proportional to the inverse of the velocity in a nonlinear way. Specifically, during the lifetime of the water drops, the velocity will be changed with a value that is nonlinearly proportional to the inverse of the soil between two points in the path. Thus, water drops on a path with less soil become faster, and the soil on the path is decreased. Changes of the soil and velocity have an influential role on the probability of selecting the direction of flow. The probability of selecting the next path is inversely proportional to the soil of the available paths. As a result, a path with low soil has a higher probability of being the selected path. The whole process will converge when the probability of selecting the shortest path equals to 1.

The following subsections describe the detailed modeling and implementation of IWD for feature selection.

3.1 Modeling of feature selection as the IWD environment

FS aims to select a subset S from the full set of features C where the knowledge represented by C is contained in S. The process of searching for the optimal subset using IWD is modeled as a complete undirected graph $G=(V, E)$, where V is number of nodes (i.e. features) connected by set of edges E. An edge represents the choice of the next feature. An edge holds an amount of soil that represents the hardness of the local path (edge between two features). A number of water drops are spreaded randomly to the set of features, where every drop is allocated with a different feature. Water drops can be used as agents that construct the solutions (population). A water drop starts to move from its source, i.e., the first allocated feature, to the next until it completes a path. A selection mechanism is required by IWD to determine the direction of the next local path, as described in section 3.2. Every water drop has a list k has a list $V_C^{IWD_k}$, which is used to record the visited features. $V_C^{IWD_k}$ is the solution k, which is constructed by the water drop IWD_k. The population is a set of solutions which are constructed by the entire water drops i.e. $T^{IWD} = \left\{ V_C^{IWD_1}, V_C^{IWD_2}, ..., V_C^{IWD_k}, ..., V_C^{IWD_{N_{IWD}}} \right\}$, where N_{IWD} is the maximum number of water drops.

In this article, RS dependency is used as the evaluation function to assess the goodness of the partial solution.

3.2 The Proposed IWDRSFS Model

In the following we present the main phases and steps of the proposed IWDRSFS model.

Initialization phase The initialization phase is used for initializing the static and dynamic parameters of the water drops and to spread the water drops on their sources.

i. **Initializing the static parameters**
   Static parameters are parameters that assume specific initial values at the beginning of the search, and they remain unchanged during the whole process. The static parameters of the proposed IWDRSFS model are:
   – $N_{\boldsymbol{IWD}}$: a set of water drops, which represents the set of solutions.
   – **Velocity updating parameters**$(a_v, b_v\, c_v)$: set of parameters used for updating the velocity of the water drops (equation 5).
   – **Soil updating parameters** $(a_s,\ b_s,\ c_s)$: set of parameters used for computing the amount of changes in the soil of the local path (equation 6).
   – **MaxIter**: the maximum number of iterations for a water drops before terminating the IDW algorithm.
   – **initSoil**: the initial value of the local soil.
ii. **Initializing the dynamic parameters**
   Dynamic parameters are parameters that are initialized at the beginning of the search, and are updated dynamically during the lifetime of search.
   – $\mathbf{V_C^{IWD_k}}$: a list of visited features for each water drop k,
   – $\mathbf{intiVel^{IWD_k}}$: the initial velocity of water drop **k** at the beginning of the search.
   – $\boldsymbol{Soil^{IWD_k}}$: the initial soil of water drop k, at the beginning of the search, where $1 \leq k \leq N_{IWD}$.
   dynamic parameters should be reset to their default initial values at the beginning of iteration.
iii. **Spread drops on their sources**
   Water drops are spread randomly to the set of features, where every drop k is allocated with a different feature, which is considered as the source of water drop. $V_C^{IWD_k}$ is updated by adding the source.

Construction phase The main goal of the construction phase for every water drop is to complete its solution starting from the source (the first point the water drop is spread on). The construction phase is completed by the fluency of all water drops amongst the features using the following four steps:

i. **Edge selection mechanism**
   A water drop k, which is resided in the current feature i can determine the next feature j, which is not in the visited list ($V_C^{IWD_k}$) using the probability

$p_i^{IWD_k}(j)$ as shown in equation (1). $V_C^{IWD_k}$) is updated by adding the selected edge.

$$p_i^{IWD_k}(j) = \frac{f(soil(i,j))}{\sum\limits_{l \notin V_C^{IWD_k}} f(soil(i,l))} \tag{1}$$

where $f(soil(i,j)) = \frac{1}{\varepsilon + g(soil(i,j))}$, ε is a small positive number prevents the division by zero in $f(.)$

$$g(soil(i,j)) = \begin{cases} soil(i,j) & if \min\limits_{\forall l \notin V_C^{IWD_k}} soil(i,l) \geqslant 0, \\ soil(i,j) - \min\limits_{\forall l \notin V_C^{IWD_k}} soil(i,l) & Otherwise. \end{cases}$$

Where $soil(i,l)$ refers to the amount of soil on the local path between features i, and j. The function min (.) returns the minimum value among all available values for its argument.

ii. **Update the velocity and local soil**
The velocity of the drop k at time $t+1$ is denoted as $vel^{IWD_k}(t+1)$. It is changed every transit from feature i to feature j using equation (2).

$$vel^{IWD_k}(t+1) = vel^{IWD_k}(t) + \frac{a_v}{b_v + c_v * soil(i,j)} \tag{2}$$

where a_v, b_v, c_v are the static parameters used to represent the non-linear relationship between the velocity of a water drop k, (i.e. vel^{IWD_k}), and the inverse of soil onto the local path, (i.e. *soil(i, j)*). *soil(i, j)* and the amount of soil carried by the drop k (i.e. $soil^{IWD_k}$) are updated by $\triangle soil(i,j)$ using equations (5), (6) respectively. $\triangle soil(i,j)$ refers to the amount of soil removed from the local path and carried by the drop. $\triangle soil(i,j)$ is nonlinearly proportional to the inverse of vel^{IWD_k} as shown in equation (3).

$$\triangle soil(i,\ j) = \frac{a_s}{b_s + c_s * time(i,j : vel^{IWD_k}(t+1))} \tag{3}$$

where, a_s, b_s, c_s are the static parameters used to represent the non-linear relationship between $\triangle soil(i,j)$ and the inverse vel^{IWD_k}. $time(i,j : vel^{IWD_k}(t+1))$ refers to the time needed for a drop k to transit from feature i to feature j at time t+1. It can be calculated using equation (4).

$$time(i,j : vel^{IWD_k}(t+1)) = \frac{HUD(i,\ j)}{vel^{IWD_k}(t+1)} \tag{4}$$

where $HUD(i,j)$ is the heuristic desirability of the edge between features i and j. In this work, the RS dependency is used to evaluate the goodness of the path between two features.

$$soil(i,j) = (1-\rho_n) * soil(i,j) - \rho_n * \triangle soil(i,j) \tag{5}$$

$$soil^{IWD_k} = soil^{IWD_k} + \triangle soil(i,\ j) \tag{6}$$

where ρ_n is a small positive constant between zero and one.

Reinforcement phase A solution with the minimum number of features amongst T^{IWD}, called the iteration best solution (i.e T^{IB}), is selected using equation (7). For each iteration, if T^{IB} is shorter than the best solution found so far, the total best solution i.e (T^{TB}) is replaced with T^{IB}. Otherwise T^{TB} is kept unchanged. To reinforce water drops in the subsequent iterations to follow T^{TB}, , the soil of all edges (i.e. the global path soil) exist in T^{IB} is updated using equation (8).

$$T^{TB} = arg \min_{\forall l \in T^{IWB}} q(x) \tag{7}$$

where $q(.)$ is the function that is used to evaluate the quality of the solutions. In feature selection, it refers to the number of features in a solution (i.e. cardinality of the solution).

$$soil(i,\ j) = (1 + \rho_{IWD}) * soil(i{,}j) - \rho_{IWD} * \frac{1}{q(T^{IWB})} \tag{8}$$

where $q(T^{IB})$ is cardinality of T^{IB}, and ρ_{IWD} is a positive constant.

Termination phase Construction and reinforcement phases are repeated until the termination criterion (i.e. the maximum number of iterations, MaxIter) is satisfied. At any iteration, if T^{IB} is better than T^{TB}, T^{TB} is replaced by T^{IB} otherwise T^{TB} is kept unchanged, as shown in equation (9). The IWD dynamic parameters are reset to their default values at the beginning of each iteration.

$$T^{TB} = \begin{cases} T^{IB} & if\, q(T^{IB}) < q(T^{TB}) \\ T^{TB} & Otherwise. \end{cases} \tag{9}$$

4 Experiments and results

The proposed IWDRSFS model was evaluated using six benchmark data sets obtained from [7], because they had been preprocessed, such as discretizing real valued features, treating the missing values, and removing outlier instances. Most of these data sets came from the UCI machine learning repository [2]. The chosen data sets had different degrees of difficulties, e.g. the numbers of features (dimensions) varied from low (13) to high (56), and the numbers of samples were small for high dimensional data sets, as shown in Table 1.

The IWDRSFS model was implemented using the Java programming language. The experiments were conducted using an Intel Pentium 4 core 2 Quad 2.66 GHz personal computer. The parameter setting of IWDRSFS is summarized in Table 2.

RS dependency was used as the evaluation function to measure the goodness of the partial solution, where a dependency of 1 was used as the stopping criterion for a complete solution.

Table 3 shows the results of IWDRSFS for the six data sets. The results of IWDRSFS are compared with those from four state-of-the-art RS methods for

Table 1. the main properties of the data sets

No.	Data sets	Abbreviations	No. of features	No. of samples
1	Artificial domains concept	M-of-N	13	1000
2	Statlog German credit data	CREDIT	20	1000
3	Letter recognition	LETTERS	25	26
4	Dermatology	DERM	34	366
5	Water Treatment Plant	WQ	38	521
6	Lung Cancer	LUNG	56	32

Table 2. IWDRSFS Parameter settings

Description	Parameters	Values
Static parameters	N_{IWD}	Number of features
	a_v, b_v, c_v	1000, 0.01, 1
	a_s, b_s, c_s	1000, 0.01, 1
	$initSoil$	100
	$MaxIter$	250
	ϵ, ρ_{IWD}, ρ_n	0.01, 0.9, 0.9
Dynamic parameters	$V_C^{IWD_k}$	Empty
	$intiVel^{IWD_k}$	4
	$soil^{IWD_k}$	0

FS, as published in [8]. They included the RS attribute reduction algorithm based on the greedy hill-climbing technique (RSAR), entropy-based data reduction (EBR), ant colony rough set attribute reduction (AntRSAR), genetic algorithm rough set attribute reduction (GenRSAR). For each data set, the experiment was repeated 20 times.

AntRSAR, GenRSAR, and IWDRSAR are multi-solution methods (i.e., every run may provide a different solution with different dimension). The results of AntRSAR, GenRSAR, and IWDRSAR in Table 3 are presented as a number with parenthesized superscript. The number refers to the dimension of the solution (i.e. a smaller subset size is better a larger subset size). The superscript refers to the number of runs that provides the corresponding dimension. On the other hand, RSAR and EBR are single solution methods, i.e. they provide the same solution even with different runs. So, the RSAR, and EBR results are presented as a single number.

As shown in Table 3, IWDRSFS outperformed RSAR and EBR in all data sets, except for CREDIT where RSAR performed better than IWDRSFS. Comparing IWDRSFS with GenRSAR; out of the six data sets; IWDRSFS found better solutions in three data sets (i.e., 4, 5, and 6), and comparable solutions in the remaining data sets (i.e. 1, 2, and 3). Comparing with AntRSAR, IWDRSFS produced comparable results for 4 data sets (i.e. 1, 3, 4, and 6).

In general; the results indicate that IDWRSFS outperforms local-based search methods (RSAR, and EBR) and are comparable with population-based search methods (AntRSAR and GenRSAR). The success of IWDRSFS for FS is owing to the characteristic of exploration, where the solutions from different places in the solution space are explored. Then, the search process is guided by the strategy that maintains the history of the search learned in the previous iterations.

5 Conclusions

This article has proposed a new FS method, i.e. IDWRSFS that combines the IDW algorithm with RS. IWD is used as the search procedure, and RS dependency is used as the subset evaluation function. IWDRSFS has been evaluated using six benchmark data sets. The empirical evaluation has shown that IWDRSFS is suitable for FS with RS, whereby good solutions have been produced. A performance comparative study with four RS-based FS methods has been carried out. The results of IWDRSFS are generally better than those from RSAR and EBR. Furthermore, the results of IWDRSFS are comparable with those from GenRSAR and AntRSAR.

While IWDRSFS has shown good results for FS, future work to improve the robustness of IWDRSFS by adapting a local search method and tuning the IWD parameters can be conducted. In addition, further investigations to verify the usefulness of IWDRSFS for real-valued data sets are required.

Table 3. :- Results of 20 runs of IWDRSFS for six UCI datasets. The results are compared with those from four state of the art methods as published in [8].

No.	Dataset	No. of Features	Comparative methods RSAR	EBR	AntRSAR	GenRSAR	IWDRSAR
1	M-of-N	13	8	6	6	$6^{(6)}7^{(12)}$	$6^{(18)}7^{(2)}$
2	CREDIT	20	9	10	$8^{(12)}9^{(4)}10^{(4)}$	$10^{(6)}11^{(14)}$	$10^{(12)}11^{(8)}$
3	LETTERS	25	9	9	8	$8^{(8)}9^{(12)}$	$8^{(16)}9^{(4)}$
4	DERM	34	7	6	$6^{(17)}7^{(3)}$	$10^{(6)}11^{(14)}$	$6^{(2)}7^{(3)}8^{(5)}9^{(5)}10^{(5)}$
5	WQ	38	14	14	$12^{(2)}13^{(7)}14^{(11)}$	16	$13^{(3)}14^{(17)}$
6	LUNG	56	4	4	4	$6^{(8)}7^{(12)}$	$4^{(6)}5^{(12)}6^{(2)}$

Acknowledgements

This research is supported by the grant with id no. $1001/PKOMP/817047$.

References

1. Abdullah, S., Jaddi, N.: Great deluge algorithm for rough set attribute reduction. Database Theory and Application, Bio-Science and Bio-Technology pp. 189–197 (2010)
2. Blake, C., Merz, C.: {UCI} repository of machine learning databases (1998), http://www.ics. uci.edu/ mlearn/MLRepository.html.

3. Dash, M., Liu, H.: Feature selection for classification. Intelligent data analysis 1(1-4), 131–156 (1997)
4. Duan, H., Liu, S., Wu, J.: Novel intelligent water drops optimization approach to single ucav smooth trajectory planning. Aerospace Science and Technology 13(8), 442–449 (2009)
5. Hedar, A., Wang, J., Fukushima, M.: Tabu search for attribute reduction in rough set theory. Soft Computing-A Fusion of Foundations, Methodologies and Applications 12(9), 909–918 (2008)
6. Hendrawan, Y., Murase, H.: Neural-intelligent water drops algorithm to select relevant textural features for developing precision irrigation system using machine vision. Computers and Electronics in Agriculture 77(2), 214–228 (2011)
7. Jensen, R.: A collection of datasets used in feature selection experimentation, http://users.aber.ac.uk/rkj/test2/ (Visited 14-10-2012)
8. Jensen, R., Shen, Q.: Finding rough set reducts with ant colony optimization. In: Proceedings of the 2003 UK workshop on computational intelligence. vol. 1 (2003)
9. Jensen, R., Shen, Q.: Computational intelligence and feature selection: rough and fuzzy approaches, vol. 8. Wiley-IEEE Press (2008)
10. Kim, H., Howland, P., Park, H.: Dimension reduction in text classification with support vector machines. Journal of Machine Learning Research 6(1), 37 (2006)
11. Niu, S., Ong, S., Nee, A.: An improved intelligent water drops algorithm for achieving optimal job-shop scheduling solutions. International Journal of Production Research 50(15), 4192–4205 (2012)
12. Pawlak, Z.: Rough sets. International Journal of Parallel Programming 11(5), 341–356 (1982)
13. Shah-Hosseini, H.: Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem. International Journal of Intelligent Computing and Cybernetics 1(2), 193–212 (2008)
14. Shah-Hosseini, H.: An approach to continuous optimization by the intelligent water drops algorithm. Procedia-Social and Behavioral Sciences 32, 224–229 (2012)
15. Shah-Hosseini, H.: Intelligent water drops algorithm for automatic multilevel thresholding of grey–level images using a modified otsu's criterion. International Journal of Modelling, Identification and Control 15(4), 241–249 (2012)
16. Shah_Hosseini, H.: Problem solving by intelligent water drops. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. pp. 3226–3231. IEEE (2007)
17. Suguna, N., Thanushkodi, K.: A novel rough set reduct algorithm for medical domain based on bee colony optimization. JOURNAL OF COMPUTING 2(6), 49–54 (2010)
18. Wang, J., Hedar, A., Zheng, G., Wang, S.: Scatter search for rough set attribute reduction. In: Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on. vol. 1, pp. 531–535. IEEE (2009)
19. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research 5, 1205–1224 (2004)