The Dimensionality Reduction Methods Based on Computational Intelligence in Problems of Object Classification and Diagnosis

Sergey A. Subbotin$^{(\boxtimes)}$ and Andrii A. Oliinyk

Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine
subbotin@zntu.edu.ua, olejnikaa@gmail.com

Abstract. The set feature selection methods based on the paradigms of computational intelligence (evolutionary search and swarm intelligence) is proposed. Proposed methods speed up the search through the creation of special operators, taking into account a priori information about the data sample and concentrating search on the most perspective solution areas. This allows preserving the stochastic nature of the search to accelerate the obtainment of acceptable solutions through the introduction of deterministic component in the search strategy. The theoretical estimates of the computational (temporal) and spatial complexity of the developed methods are obtained. The proposed methods are experimentally studied on a set of problems of automatic object classification, technical and medical diagnosis. On the results of experiments the comparative characteristics and recommendations for the use of the proposed methods are given.

Keywords: Dimensionality reduction · Diagnosis · Classification · Feature selection

1 Introduction

The traditional tool for data sample dimensionality reduction is a feature selection methods [1–3], that select from a given set of input features a subset of the most significant features for computing the output feature value. According to the type of procedure of new solution formation on the basis of existing decisions they are divided on exhaustive search methods [1, 4] (implement deterministic transition from one decision to another, are highly complex and time-consuming) and stochastic search methods [2, 5] (implement random transition from the current to new solutions, but probabilistically taking into account information about previous solutions, avoid the falling into local minima, and find suboptimal solutions for a limited time).

The usage of known exhaustive methods [1, 4, 6] requires a large number of evaluations of feature combinations, produced from the initial set of features, which makes impossible to use this approach for a big number of features in the initial set, since it has a huge computational cost. Therefore, for feature selection it is reasonable to use the intelligent stochastic search methods [2, 3, 7], because they are more suited to finding new solutions by combining the best solutions obtained at different iterations

and does not require a search of all solutions. However, known intelligent search methods are slow because they do not use the problem specific information in the process of a search.

Therefore, the aim of this work is to create a complex of Computational Intelligence methods for solving problems of feature selection making possible to accelerate the obtaining of solutions.

2   Formal Problem Statement

Let $<x, y>$ is a sample of precedents (cases, instances, exemplars), where $x = \{x^s\}$, $x = \{x_j\}$, $x^s = \{x_j^s\}$, $x_j = \{x_j^s\}$, $y = \{y^s\}$, $s = 1, 2, \ldots, S, j = 1, 2, \ldots, N; x_j^s$ is a value of j-th input feature x_j of exemplar x^s, y^s is an output for precedent x^s (for the classification problems $y^s \in \{1, 2, \ldots, K\}$, $K > 1$, K is the number of classes), S is a number of precedents, N is a number of input features. Denote as $<x', y'>$ a fragment of $<x, y>$, as N' a number of features in $<x', y'>$, as $f()$ a user criterion describing the quality of argument according to the decided task, as opt an optimal desirable or acceptable value of the functional $f()$ for the problem. Then the problem of feature selection is for given $<x, y>$ find $<x', y'>$, $x' \subset \{x_j\}$, $N' < N$, $S' = S$, $f(<x', y>, <x, y>) \rightarrow opt$.

3   Background of Computational Intelligence Based Search

Among the stochastic search methods the wide usage have evolutionary [3, 7, 8] and multi-agent [9–11] methods.

The evolutionary methods based on the idea of the evolution of solutions through simulation of natural selection. At the beginning the epoch counter setted as $t = 0$ and method form an initial population of solutions $P_t = \{H_j\}$, where $j = 1, 2, \ldots, N$, N is a number of population solutions, $H_j = \{h_{ij}\}$ is a j-th solution, h_{ij} is a value of i-th bit (gene) of j-th solution, $i = 1, 2, \ldots, L$, where L is a length of the solution. Then performed a cyclic replacement of one population solutions by the next more adapted. For this the solutions fitness is estimated for current t-th population $\{f(H_j)\}$. Then the checking of search stop conditions (reaching the epochs limit or a reasonable value of the objective function) is performed. If the conditions are satisfied then the search will stop, and otherwise for the current population of solutions method execute selection operator (on the basis of $f(H_j)$ values it probabilistically select solutions for crossover and mutation, with a preference for more adapted solutions), crossover operator (forming parent pairs from which create new solutions) and mutation operator (made spontaneous changes in decisions to increase the diversity of solutions).

On the basis of produced solutions a new generation of is formed, which includes new solutions, as well as the best solutions, the objective function values of which are the best in the population. Crossover of the fittest solutions results in that the most perspective search space parts are explored. Finally, the population of solutions will converge to an optimal task solution.

Multi-agent methods [3, 9–11] is a group of multi-dimensional stochastic search methods based on modeling the behavior of self-organizing agent colonies having the

collective nature and provided by coordination (controls the spatial-temporal agent placement), specialization (distributes agent actions and form the spatial, temporal and social relations arising in the colony's work), cooperation (agents association mechanisms) and collective decision-making of agents (colony allows agents to react on changes in the external environment). These methods includes indirect agent communication methods (ant colony optimization (ACO) [3, 7], particle swarm optimization (PSO) [11], bacteria foraging optimization (BFO) [9] and direct communication methods (Bee Colony Optimization (BCO) [10]).

The methods of stochastic search are able to optimize the functions of any complexity without additional requirements on the objective function (unimodality, continuity, smoothness, monotonicity, differentiability). Thus, the use of stochastic search techniques is appropriate in the problems of feature and sample selection.

Despite these advantages the intelligent methods based on stochastic search has such common disadvantages as high iteratively and dependability of the methods speed from the choice of the search starting point (usually defined randomly) and a priori uncertainty of the time of convergence of methods due to the stochastic nature of the search, as well as neglect of search operators in available information from sample. Therefore, the improvement of the speed of intelligent stochastic search methods by reducing these disadvantages is urgent.

4 The Feature Selection Based on the Intelligent Stochastic Search

To speed up the feature selection based on stochastic search we propose to retain their stochastic nature add them deterministic components by modifying the search operators, under which they will take into account a priori information about the data sample. As such information we propose to use the individual evaluations of feature informativity [3, 11]. Since proposed methods are based on classical methods of intelligent stochastic search further we shall only present their key differences from the known basic methods.

4.1 Feature Selection Method Using the Entropy

This method as a search model use a canonical evolutionary search [8] with improvements listed below. Solution forming stage: set the probability of inclusion of i-th feature to the solution: $P_i = 1 - e_i$, where e_i – entropy [9, 10] of i-th feature. Solution crossover stage: use mask $h^* = \{h_i^*\}$, where $h_i^* = 1$, if $e_i < e_{\text{п.}}$ ($e_{\text{п.}}$ – threshold), $h_i^* = 0$, otherwise. Solution mutation stage: the mutation probabilities P_M of features with high e_i are increased, and with low e_i are decreased. Solution selection stage: the goal function is defined as

$$f\left(H_j\right) = \left(1 + \left(\sum_{i=1}^{N} h_{ij}\right)^{-1} \left(\sum_{i=1}^{N}\sum_{k=1}^{N} h_{ij}h_{ik}d_{ik}\right) E_j\right)^{-1} \left(\sum_{i=1}^{N} e_i \cdot h_{ij}\right)^{-1}, \; E_j \geq 0, \quad (1)$$

where H_j is a j-th solution, h_{ij} is an i-th digit of j-th solution, E_j is a model error for j-th solution, d_{ik} is an individual informativity value of i-th feature relative to k-th feature.

4.2 Feature Selection Method with Feature Grouping

This method as a search model use a canonical evolutionary search [3, 8] with improvements listed below. Solution forming stage: the set of solutions $\{H_j\}$, $H_j = \{h_{ij}\}$ is formed, where h_{ij} is a digit specifying usage of i-th feature taking into account its individual informativity value I_i. Solution crossover stage: using mask $h^* = \{h_i^*\}$, where $h_i^* = 1$, if $I_i > I_{tr.}$, ($I_{tr.}$ is a previously specified threshold value), $h_i^* = 0$, otherwise. Solution mutation stage: the mutation probabilities P_M of features with high I_i are decreased, and with low I_i are increased. Solution selection stage: the goal function is defined as:

$$f(H_j) = \left(\sum_{i=1}^{N} I_i h_{ij} \right) \left(1 + \left(\sum_{i=1}^{H} h_{ij} \right) \left(\sum_{i=1}^{N} \sum_{k=1}^{N} h_{ij} h_{ik} d_{jk} \right) E_j \right)^{-1}, E_j \geq 0. \quad (2)$$

4.3 Evolutionary Feature Selection Method with Clustering

This method as a search model use a canonical evolutionary search [3, 8] with improvements listed below.

Initialization stage: the individual feature informativity values $\{I_i\}$ determined relatively to outputs, and the feature relations indicators are determined. The similar features are grouped. The most informative feature in each group is selected, that is a center of a cluster C_q.

Solution forming stage: the probability of i-th feature inclusion to the solution is determined on the Euclidean distance basis from it to the cluster center $d(x_i, C_q)$, also as the feature and cluster center individual informativities I_i and I_{Cq}, then $P_i = I_i + (I_i - I_{Cq}) \{d(x_i, C_q)/d_{qmax} \mid x_i \in C_q, q = 1, 2, \ldots, Q\}$, where d_{qmax} is a maximal distance in q-th cluster.

4.4 Feature Selection with Fixing of the Search Space Part

This method at initialization stage order features by I_i growing, then first $\alpha \eta N$ features are deleted, where α is a given coefficient, $0 < \alpha \leq 1/\eta$, η is a search space reduction coefficient:

$$\eta = N^{-1} \sum_{i=1}^{N} \{1 \mid I_i < I_{avg.}\}, \quad (3)$$

where $I_{avg.}$ is an average feature informativity value. Then method perform canonical evolutionary search for reduced feature set.

4.5 Multi-agent Feature Selection Method with Representation of the Destination Points by Features

This method as a search model use a multi-agent search with indirect communication on the basis of ACO model [3, 13] with such improvements. Search space forming: vertices of the search graph (destination points) represented by features. Search strategy: agent must pass the way to the specified number of destinations N', which determines the number of features that must be leave. Solutions encoding: path traveled by the j-th agent is a feature set (decision) H_j, on which the model is constructed: $H_j = \{h_{ij}\}$, $h_{ij} = 1$, if the point i is included in the j-th agent path, $h_{ij} = 0$ – otherwise.

4.6 Multi-agent Feature Selection Method Based on BFO Model

This method as a search model use multi-agent search with indirect agent communication based BFO model [9] with following improvements. Search space: each point in a search space represented by binary string, which consists of feature informativities as digits. Search strategy: the BFO model is hybridysed by using evolutionary operators of proportional selection and simple mutation. Search speed increasing: at the initialization stage for each i-th agent, $i = 1, 2, ..., H$, set initial position: $h_{ji} = 1$, if $rand_{ji} < I_j \left(\sum_{q=1}^{N} I_q \right)^{-1}$, $h_{ji} = 0$, otherwise, where $rand_{ji} \in [0; 1]$ is random numbers, I_j is an individual informativity value of j-th feature, $j = 1, 2, …, N$.

4.7 Multi-agent Feature Selection Method with Representation of the Destination Points by Feature Informativities

This method as a search model use a multi-agent search with indirect communication on the basis of ACO model [3, 13] with listed below enhancements. Search space forming: search graph vertices (destination points) represented by randomly formed binary set $B = \{b_i\}$, $b_i = \{0;1\}$, $i = 1, 2, …, N$, which contains N' element equal to one. Search strategy: each agent must pass the way to all destination points. Solutions encoding: the path traveled by the j-th agent by destination points is a binary set of informative features (decision) B used for model construction.

4.8 Multi-agent Feature Selection Method with Direct Agent Communication

This method as a search model use a multi-agent search with direct agent communication based BCO model [10, 14] with such improvements. Search space: $(N' \times N)$ – dimensional space, the feature informativity is a resource collected by the agent. Solution evaluation: based on agent stay on the iteration t in source h, in which $H^h(t)$ agents are located: $l^h(t) = a_h/H^h(t)$, $h = 1, 2, …, N' \times N$, where $a_h = \varepsilon^*/E_h$, E_h is a model error for source h, ε^* is a reduction factor. Search speed increasing: when the agents-scouts starts the one part of scouts randomly placed in the search space

(a condition for global search), and the other part is placed in the sources of resources proportionally the values of their informativity (accounting a priori information).

5 Complexity Analysis

For the developed feature selection methods at the sequential implementation of computations we obtain analytical assessments of temporal (computational) and spatial complexities (Table 1). Table 1 use the following notation: H is a number of agents (for multi-agent search) or solutions (for the evolutionary search), F is a complexity of the simulation of neuro-fuzzy network based model and the complexity of the error calculation on its base for the learning sample, T is a number of iterations taken by the method for search. The precise estimates of the complexity in the Table 1 are given in the soft form (without suppressing of members with less powers by members with greater powers). Taking $n = NS \approx N^2$ we present rough estimates in hard form (members with greater powers suppress members with a lesser powers).

Table 1. Estimates of the complexity of feature selection methods based on stochastic search

Method	Time complexity		Spatial complexity	
	Accurate	Rough	Accurate	Rough
4.1	$O(12NS + N^2 + N + THF)$	$O(n^2\sqrt{n})$	$O(13NS + 0.0625HFN)$	$O(n^2)$
4.2	$O(12SN(N + 1) + N + THF)$	$O(n^2\sqrt{n})$	$O(N(S + N + 2)$ $+ 0.0625HFN)$	$O(n^2)$
4.3	$O(12NS + N + THF)$	$O(n^2\sqrt{n})$	$O(NS + 2N + 0.0625HFN)$	$O(n^2)$
4.4	$O(1.5S(N - 1) +$ $C + 5N + TFH)$	$O(n^2\sqrt{n})$	$O(NS + 3N +$ $0.0625HFN + C)$	$O(n^2)$
4.5	$O(TH(N^2 + 4N + H + F))$	$O(n^2\sqrt{n})$	$O(NS + H(3N + F + H))$	$O(n\sqrt{n})$
4.6	$O(TH(N^2 + F + 4N))$	$O(n^2\sqrt{n})$	$O(NS + 3HN + HF)$	$O(n\sqrt{n})$
4.7	$O(8H^2 + 6H^2N + HF + N +$ $TH(2F + H + N + 4))$	$O(n^2\sqrt{n})$	$O(NS + HN + 3HF)$	$O(n\sqrt{n})$
4.8	$O(H + TH(F + 9 + HN - H))$	$O(n^2\sqrt{n})$	$O(NS + 5H + HN^2 -$ $HN + FH)$	$O(n\sqrt{n})$

6 Experiments and Results

To study the proposed methods they have been implemented as software. The experimental investigation of the proposed methods and software was performed by solving practical problems, which sample characteristics shown in Table 2.

The results of experimental study are shown in Table 3. Here t is a time of feature selection, Nm is a computer memory volume used by the method, N'/N is a the proportion of selected features in original feature set.

Our experiments confirmed the efficiency of the developed methods. The proposed feature selection methods allow to significantly increase the speed of work, compared with the canonical evolutionary search [3, 8], because they use fewer references to the objective function and, consequently, require fewer model constructs.

The proposed methods based on evolutionary search can be recommended for use in cases where there are no significant limitations on the available computer memory, and methods based on multi-agent search – when computer memory is limited.

Table 2. The practical tasks collection for experiments

Task	Code	N	S
Diagnosis of air-engine blades [3]	SIGNAL	10240	32
Simulation the total index of quality of life of patients sick of a chronic obstructive bronchitis [11]	KOG	106	86
Diagnosis of chronic obstructive bronchitis [12]	HBR	28	205
Modeling of dependence of state of children's health from environmental pollution [11]	ECO	43	954
Automatic classification of vehicles [11]	AUTO	26	1062
Automatic classification of agricultural plants [11]	PLNT	55	248

Table 3. Averaged experimental performance evaluation of feature selection methods

Task	Canonical evolutionary method			Method with fixing of the search space part			Method with feature grouping		
	t, sec	N_m, M	N'/N	t, sec	N_m, M	N'/N	t, sec	N_m, M	N'/N
SIGNAL	10317	7.22	0.113	3317.6	8.19	0.115	3317.6	8.19	0.115
KOG	21909	10.00	0.132	6984.1	10.10	0.132	6984.1	10.10	0.132
HBR	17150	14.12	0.786	5429.2	14.75	0.679	5429.2	14.75	0.679
ECO	43335	430.40	0.651	13832	487.84	0.628	13832	487.84	0.628
AUTO	35784	344.43	0.500	10518	373.90	0.500	10518	373.90	0.500
PLNT	4714	40.74	0.200	1538.2	45.09	0.218	1538.2	45.09	0.218

Task	Method using entropy		Method with feature clustering			BCO based method		
	t, sec	N_m, M	t, sec	t, sec	N_m, M	N'/N	t, sec	N_m, M	N'/N
SIGNAL	2947.1	7.73	0.113	2853	7.74	0.115	6008.5	3.06	0.115
KOG	5864	10.65	0.113	6686.6	9.55	0.132	13558	3.73	0.113
HBR	4700.1	14.46	0.714	4591.6	13.86	0.750	9822.1	5.30	0.750
ECO	12867	500.39	0.651	14764	472.47	0.628	28690	168.21	0.628
AUTO	10065	364.00	0.500	10400	355.00	0.577	22164	129.84	0.577
PLNT	1334.2	45.94	0.200	1436.6	41.83	0.164	3072.3	15.91	0.200

Task	ACO with features			ACO with feature informativities			BFO based		
	t, sec	N_m, M	N'/N	t, sec	N_m, M	N'/N	t, sec	N_m, M	N'/N
SIGNAL	5748.5	0.89	0.115	5855.3	0.93	0.111	5946.2	1.21	0.113
KOG	12282	1.34	0.123	12792	1.25	0.132	14033	1.66	0.142
HBR	9327.3	1.96	0.750	9739.1	1.72	0.786	9394.1	2.28	0.714
ECO	28000	57.60	0.651	25507	57.30	0.651	27119	76.35	0.651
AUTO	19693	43.18	0.500	19378	43.89	0.538	21036	56.94	0.538
PLNT	2922.6	4.98	0.164	2814.9	4.68	0.200	2894.8	6.52	0.200

7 Conclusion

The problem of data dimensionality reduction have been studied. The dimensionality reduction methods based on intelligent stochastic search including feature selection methods and sample selection methods have been proposed. The experiments on study of proposed methods are conducted. They show that proposed set of methods allow to significantly reduce the data dimensionality.

References

1. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. **1**, 131–156 (1997)
2. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. John Wiley & Sons, Hoboken (2008)
3. Boguslayev, A.V., Oleynik, A.A., Oleynik A.A. et al.: Progressivnyye tekhnologii modelirovaniya, optimizatsii i intellektual'noy avtomatizatsii etapov zhiznennogo tsikla aviatsionnykh dvigateley: monografiya. Motor Sich, Zaporozhye (2009). (in Russian)
4. Subbotin, S.A.: Methods of sampling based on exhaustive and evolutionary search. Autom. Control Comput. Sci. **3**(47), 113–121 (2013)
5. Korobiichuk, I., Podchashinskiy, Y., Shapovalova, O., Shadura, V., Nowicki, M., Szewczyk, R.: Precision increase in automated digital image measurement systems of geometric values. In: Jabłoński, R., Brezina, T. (eds.) Advanced Mechatronics Solutions. Advances in Intelligent Systems and Computing, vol. 393, pp. 335–340. Springer, Heidelberg (2016). doi:10.1007/978-3-319-23923-1_51
6. Korobiichuk, I., Bezvesilna, O., Ilchenko, A., Shadura, V., Nowicki, M., Szewczyk, R.: A mathematical model of the thermo-anemometric flowmeter. Sensors **15**, 22899–22913 (2015)
7. Engelbrecht, A.: Computational intelligence: an introduction. Wiley, Sidney (2007)
8. Ruan, D.: Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms. Springer, Berlin (2012)
9. Liu, Y., Passino, K.M.: Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. J. Optim. Theory Appl. **3**, 603–628 (2002)
10. Karaboga, D., Akay, B.: A survey: algorithms simulating bee swarm intelligence. Artif. Intell. Rev. **31**, 61–85 (2009)
11. Subbotin, S.A., Oleynik, A.A., Gofman, Y.A. et al.: Intellektual'nyye informatsionnyye tekhnologii proyektirovaniya avtomatizirovannykh sistem diagnostirovaniya i raspoznavaniya obrazov: monografiya. SMIT Co., Kharkov (2012)
12. Subbotin, S., Oleynik, A.: Entropy based evolutionary search for feature selection. In: Proceedings 9th International Conference (CADSM-2007), The Experience of Designing and Application of CAD Systems in Microelectronics, pp. 442–443. IEEE Press, Lviv (2007)

13. Subbotin, S., Oleynik, A.: Modifications of ant colony optimization method for feature selection. In: Proceedings 9th International Conference (CADSM-2007), The Experience of Designing and Application of CAD Systems in Microelectronics, pp. 493–494. IEEE Press, Lviv (2007)
14. Oliinyk, A.O., Oliinyk, O.O., Subbotin, S.A.: Agent technologies for feature selection. Cybern. Syst. Anal. **2**(48), 257–267 (2012)