Description of the Method

EM-like metaheuristic for dimensionality reduction is designed as EM optimization method, with some specific aspects that depends on dimensionality reduction problem (Kartelj 2015). Four aspects that make difference between EM for dimensionality reduction and other EM method will be explained in detail.

First of distinctive characteristic is calculation of **objective function**: all EM points $\boldsymbol{p_i}, i = 1, \dots M$ are converted from the real-valued vectors to corresponding binary vectors $\boldsymbol{s_i}, i = 1, \dots M$ in the following way: $s_i^k = \begin{cases} 0, & p_i^k < 0.5 \\ 1, & p_i^k \geq 0.5 \end{cases}$, where 1/0 denotes if the feature is included or not. The objective function is calculated in two ways depending on the type of classifier:

(a) if 1-NN classifier is used, a 5-fold cross validation is performed and the objective value is calculated as balanced classification accuracy;

(b) if SVM classifier is employed, 2-fold cross-validation is performed, while the objective value is calculated as classification accuracy, i.e. the average percentage of correctly predicted records.

Second distinctive characteristic is using **local search** (LS) procedure. Upon calculation of objective value for all EM points, local search procedure is applied to at most one of them. At the beginning, EM for dimensionality reduction examines if a local search (LS) procedure is to be called, or not. The reasons for this examination are twofold:

(a) LS procedures are usually time consuming, so any reduction of LS calls can (often significantly) decrease the algorithm execution time;

(2) precisely tuned criteria whether the LS is or is not called can increase the exploratory properties of the candidate solution space, directing the search process into the promising unexplored regions.

Therefore, in this EM method, the LS procedure is called only if the conjunction of the following conditions is satisfied:

(a) The EM point has the best, or the second best objective value;

(b) LS have not ever been applied on the current EM point, or its objective value has been changed since the last application of LS;

(c) The best objective value has not been changed for at least 10 generations;

After that, if the criteria for calling the LS on the current point are satisfied, the LS is performed. LS consist of two procedures: the first one is 1-swap LS with immediate application of improvement, and the second is 2-swap LS with application of the best found improvement.

In the ***1-swap*** procedure, a single bit is changed, including/excluding the corresponding feature in/from the solution. If the improvement is detected, it is immediately applied and the LS continue with the new, improved point. Procedure 1-swap stops when no new improvement can be found.

The second ***2-swap*** procedure, consists of the following steps: firstly, the algorithm excludes from the solution a certain feature that belongs to the solution; after that, the goal is to search for the feature that is not in the solution, which when included produces the improvement. When all pairs formed of the excluded feature and previously not included features are checked for improvement, the pair of features offering the best improvement is selected, and the swap is performed. Procedure 2-swap stops when all features belonging to the solution have been tried for exclusion.

Third distinctive characteristic is **scaling**: when the EM point coordinates (related to the features) have values close to 0.5, unnecessary dispersion of the search can appear, because crossing of the threshold value occurs too frequently. In such case, the decision whether the corresponding features are or are not included in the solution changes too often, the convergence of the algorithm is weakened and the overall searching process becomes unreliable.

In order to avoid this appearance, the standard EM method is extended by introducing the specific scaling procedure which enables a better control of the movements of EM points: $\boldsymbol{p_i} = \alpha\, \boldsymbol{s_i} + (1 - \alpha)\, \boldsymbol{p_i}$, where scaling factor α is a number between 0 and 1.

Fourth distinctive characteristic is **cashing**: prior to the calculation of objective function, vector $\boldsymbol{s_i}$ is looked up in cache collection structure. Only if it is not found, the 1-NN or SVM classifier is called to calculate the corresponding objective function. Otherwise, the objective value is taken from the cache collection.

Previously described method will be compared with various algorithms for dimensionality reduction known from the literature: EM algorithm proposed in (Chao-Ton and Hung-Chun 2011), genetic algorithm from (Yang and Honavar 1998) and with two variants of particle swarm optimization method that are proposed in (Li-Fei et al. 2012).

Computational experiments

Each of the following subsections contains results for one of the considered problems: dimensionality reduction, SVM parameter selection and maximum betweeness. Results are obtained by executing experiments on test instances that have biomedical/biological origin (various classification data sets from UCI repository and data extracted from RHMAPPER software package).

The method is implemented in C programming language, and compiled with Visual Studio 2010 compiler. All the tests are executed on PC with 2.4GHz Intel processor and 4GB RAM under Windows 7 operating system.

Two experiments were conducted, on two separate collections of classification data sets with biological/biomedicine origin taken from UCI machine learning repository (Lichman 2013).

The first collection, consisting of 6 data sets (described in Table 1), is used for comparison. Structure of that table is as follows: dimensionality (e.g. number of features – denoted as N), number of classes (N_c) and data set size (N_r). Previously described EM for dimensionality reduction is compared with EM algorithm (Chao-Ton and Hung-Chun 2011) - denoted as EM^{stc} and with genetic algorithm (Yang and Honavar 1998) - denoted as *GA*.

The second collection contains 3 data sets and it is used for comparison with two variants of particle swarm optimization method (Li-Fei et al. 2012) - denoted by PSO^1 and PSO^2.

The first experiment uses the percentage of removed features, while the second uses the complementary measure (the number of retained features) for measuring quality of the obtained solution. The objective functions are calculated by using cross validation technique. In both experiments, measure of quality is denoted as *RF*.

Experiment 1: In this experiment, in order to compare method EM with EM^{stc} and GA, accuracy and feature reduction are calculated on 1NN classifier. For each data set, the EM algorithm is executed 10 times. Every execution uses different random seed that consequently produces different fold partitioning. For each data set, the average values of RF are recorded. The number of iterations N_{it} and the number of EM points M are kept uniform across all data ssts: $N_{it} = 600$, $M = 150$. The parameter α which controls the scaling intensity is set to 0.1.

Table 2 shows comparison results. The first two columns of the table are data set name and average values obtained by *EM^{stc}* and *GA* (measured in percentages). The next column *FS* shows the average optimal solution obtained by the full search algorithm after 10 executions. The column *EM* takes value opt if the obtained average solution is equal to the average optimal solution of FS, which means that in such cases EM obtains optimal solutions in every of 10 executions. The last three columns give the reduction rate (in percentages) for three algorithms that are compared.

The results in Table 2 suggest that EM outperforms other methods on 4 out of 6 cases in terms of feature reduction rate. It can be also seen that EM obtained optimal average solution in 5 out of 6 cases, meaning that the success rate was 83.3% for these data sets.

Table 3 shows the execution times (in seconds) of the compared methods. It can be noted that 5 data sets are solved easily with FS, all in less than 10 seconds. The studied EM showed to be similarly fast on these small data sets, which is the consequence of caching.

Experiment 2: In the second experiment, the EM method is compared to PSO^1 and PSO^2 methods. The comparison is based on three data sets from UCI repository. LIBSVM implementation of SVM (Chih-Chung and Lin 2011) is used as an underlying classification algorithm. Error cost parameter of SVM is set to 100 and radial basis function with $\sigma = 2$ is used as a kernel. For multiclass data sets, one-against-rest strategy is used. The maximal number of EM iterations is set to 300. As a resampling technique, 2-fold cross validation is used. The reported values of classification accuracy and percentages of retained features are all reported as averages on running the EM algorithm 10 times for each data set.

The comparison is based on the average classification accuracy and the average percentage of retained features.

Table 4 shows the following information: data set name, number of features (N), number of classes (N_c), classification accuracy of first and second variant of PSO algorithm (PSO^1 and PSO^2), optimal classification accuracy obtained by full search algorithm (if it finishes execution - *FS*), EM classification accuracy (*EM*), and finally, average numbers of retained features (PSO^1_d , PSO^2_d and EM_d).

EM reached all average optimal solutions on smaller data sets where the FS algorithm finished its execution. In the remaining large data set, FS algorithm execution lasted more than 3 days, so it was terminated before completion. For measure number of retained features is favorable to have smaller value. It can be noted that EM reached the smallest average number of features in all data sets.

Results

data set	***N***	***N_c***	***N_r***
abalone	8	11	3842
iris	4	3	150
water	38	4	513
wine	13	3	178
wisconsin	9	2	683
yeast	8	9	1479

Table 1 - Classification data sets used in the experiment

data set	$\boldsymbol{EM^{stc}}$	***GA***	***FS***	***EM***	$\boldsymbol{EM_{RF}^{stc}}$	$\boldsymbol{GA_{RF}}$	$\boldsymbol{EM_{RF}}$
abalone	24.35	24.37	23.99	***opt***	52.50	50.00	**57.50**
iris	98.00	98.00	99.39	***opt***	55.00	**60.00**	50.00
water	73.34	66.28	-	**80.03**	54.21	47.89	**63.16**
wine	98.57	98.57	99.80	***opt***	58.46	61.54	**72.31**
wisconsin	98.25	98.04	98.62	***opt***	**53.33**	40.00	48.89
yeast	47.07	47.03	51.15	***opt***	17.50	12.50	**22.50**

Table 2 - Accuracy and feature reduction comparison for 1NN classifier

data set	***FS_t (s)***	***EM^{stc}_t (s)***	***GA_t (s)***	***EM_t (s)***
abalone	8.4	1376.1	7097.2	**7.2**
iris	**0.0**	7.5	288.6	0.9
water	>3 days	262.8	1574.5	**55.7**
wine	**1.9**	153.0	269.9	2.5
wisconsin	**0.6**	70.6	2096.8	2.0
yeast	**1.2**	234.7	2252.3	2.4

Table 3 - Computational times for 1NN classifier

data set	***N***	***N_c***	***PSO^1***	***PSO^2***	***FS***	***EM***	***PSO^1_d***	***PSO^2_d***	***EM_d***
breast-cancer	30	2	96.83	**97.66**	-	95.92	11.1	12.2	**6.4**
heart	13	2	84.30	86.01	83.74	**opt**	8.6	7.5	**3.5**
wine	13	3	99.19	99.72	97.30	**opt**	8.3	8.6	**6.7**

Table 4 - Accuracy and number of features comparison table for SVM classifier