Unsupervised feature selection algorithms for Wireless Sensor Networks

C. Alippi, G. Baroni, A. Bersani, M. Roveri
Dipartimento di Elettronica e Informazione
Politecnico di Milano, Milano, Italy
{alippi, roveri}@elet.polimi.it - {gabriele.baroni, andrea.bersani}@mail.polimi.it

Abstract — A wireless sensor network (WSN) is a distributed measurement system deployed over a geographical area to acquire physical information which, depending on the nature of the monitoring phenomenon, can be spatially correlated in space and time. Spatial correlation, to be intended here at different levels, can be exploited to reduce the communication bandwidth, implement articulated sensing and carry out energy saving policies. The paper aims at investigating unsupervised feature selection algorithms and how they can be used to exploit spatial correlation in WSNs. The interest is due to the fact that generation of a reduced set of features (i.e., aggregated data) has a positive effect on optimal energy management, hierarchical decision making and performance. Six algorithms have been critically discussed and contrasted both at theoretical and experimental levels.

Keywords-component: Wireless Sensor Networks, Unsupervised feature selection, distributed monitoring system.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are distributed measurement systems which consist of a large number of nodes deployed over a geographical area [1][2]. Each node is a low-power device that embeds sensing, processing and communication abilities. Acquired data are locally processed and transmitted through the network to a sink for further processing and data interpretation (e.g., a control room). To achieve the performance required by a distributed measurement system (e.g., event detection, monitoring, forecasting), nodes strictly cooperate, with the cooperation among nodes strongly limited by energy, processing and communication constraints [3].

Recently, a novel trend within the WSNs community sees samples more than a datastream to be conveyed to the control room. In fact, once suitably extracted from data and processed, features can be used to improve the efficacy and efficiency of the network (e.g., they constitute the basic elements to generate hierarchical event triggering, and distributed decision making). This feature-centric approach is well justified by observing that most of physical phenomena satisfy the locality properties: temporal locality (samples are related in time) and spatial locality (close views of the phenomenon provide related data). From the 'Tobler's first law of geography [4] we can even say that in geo-referenced data “everything is related to everything, but near things are more related than distant things”.

In this direction only few results have been published in the WSNs arena. For instance, [5] suggests a theoretical framework exploiting spatial and temporal correlations in data to be used for developing effective routing algorithms. Among others hypotheses, the physical phenomenon is assumed to be ruled by a Gaussian distribution with known variance.

A correlation-based compression technique for routing algorithms is proposed in [6] which relies on an empirical approximation of the joint entropy between nodes w.r.t. their distance. [7] shows a Medium Access Control layer which exploits the spatial correlation framework of [5] to reduce the number of transmitted data. A different approach is proposed in [8], which suggests exploiting the correlation among subsequent frames in wireless multimedia sensor networks to reduce the amount of data to be transmitted and stored. Unfortunately, all above solutions assume strong hypotheses and require a priori knowledge about the physical phenomenon under monitoring. However, for all of them the generation of features and their selection represents the key issue of the analysis.

The ability to measure the “mutual affinity” among spatial measurements, i.e., cross-dependency or correlation, would allow researchers for defining a new generation of solutions to routing, decision, and energy management (see Figure 1).

In fact, information-based routing algorithms could generate clusters by grouping units according to the mutual affinity of features. Instead, decision making algorithms could exploit the mutual affinity within a cluster to make decisions, e.g., detecting the presence of an event by relying on cluster-based features. Finally, an effective energy management policy
for units would exploit mutual affinity to switch off those units or sensors whose information-content is provided indirectly by others.

Moreover, mutual affinity would allow us for running distributed change detection tests (e.g., [9]) to identify changes in the physical phenomenon at the cluster and at the network level as well as detect the presence of faults affecting sensors and units.

The structure of the paper is as follows. Section II provides a theoretical review of the most relevant unsupervised feature selection algorithms; Section III and IV summarize and compare results of a wide experimental campaign leaving to Section V discussions and conclusions.

II. UNSUPERVISED FEATURE SELECTION ALGORITHMS

This section presents a critical review of the main unsupervised feature selection algorithms by discussing their characteristics, pros and cons and adaptability to WSNs applications.

A. Ranking a Random Feature for Variable and Feature Selection (SFFSA) [10]

The unsupervised feature selection algorithm presented in [10] aims at ranking the features according to the relevance w.r.t. a reference output and selecting the most relevant ones. The algorithm starts by selecting the feature that has the smallest angle with a reference output \( y^p \), according to the squared cosine figure of merit

\[
\cos^2(\theta_k, y^p) = \frac{(x_k \cdot y^p)^2}{\|x_k\|^2 \|y^p\|^2}, \quad k = 1, ..., N
\]

where \( x_k \) and \( N \) are the \( k \)-th feature vector and the number of available features, respectively. Then, the algorithm computes the value of the cumulative distribution function of the squared cosine of the angle between the selected feature and the reference output. If this value is greater than a user-defined threshold, the feature is discarded and the procedure terminates; otherwise the feature is selected and the algorithm proceeds to the next iterations.

The proposed algorithm is well supported by a strong theoretical base and is fast (its computational complexity is linear w.r.t. \( N \)). Unfortunately, it is semi-supervised since it requires a reference vector that represents the reference output. Moreover, the value of the threshold is application-dependent.

B. Streamwise Feature Selection (SFS) [11]

Differently from the previous approach, the Streamwise Feature Selection algorithm considers candidate features which are presented sequentially to the selection engine (the feature set does not need to be available in advance). For each candidate feature the algorithm computes the following test [12]: if the reduction in the minimum description length [13] provided by the inclusion of the candidate feature into the current feature set is larger than a user-defined threshold, the feature is selected; otherwise it is discarded (an equivalent statistical version of this criterion, called Alpha-investing, is also present in the paper). The algorithm then proceeds to the next candidate feature and, if the candidate feature has been selected, the threshold for adding new features is increased.

The proposed algorithm is fast and able to discard features with constant values but is not effective in case of less than 10 features. Moreover, selection of the algorithm’s thresholds is a critical issue.

C. Evolutionary Local Selection Algorithm (ELSA) [15]

The algorithm, which relies on an evolutionary local selection algorithm originally developed for artificial life models in ecological environments, aims at exploring the space of feature subsets trying to optimize a multi-objective function based on inter- and intra-cluster distance, number of clusters and number of selected features. Unfortunately, an explicit relationship among these figures of merit is not present in the paper and the optimization criterion reduces to the minimization of the intra-cluster distance.

Moreover, the algorithm, based on genetic algorithms, suffers from an excessive computational complexity and is very sensitive to the initialization of the initial population of the genetic algorithm.

D. Nondominated Sorting Genetic Algorithm (NSGA) [16]

NSGA represents an approach similar to ELSA in which a multi-objective genetic algorithm explores the feature subset space aiming at minimizing the number of features while maximizing the normalized ratio between the intra-cluster and the inter-cluster separation. The convergence criterion of the algorithm is the maximum number of iterations of the genetic algorithm. Moreover, the initialization of the algorithm’s parameters is very critical.

E. Neuro-Fuzzy Feature Selection (NNFS) [17]

The algorithm relies on the Fuzzy Feature Evaluation Index (FFEI) which is a figure of merit that measures the similarity between the samples in the original and in the reduced space for a set of transformed features. In particular, FFEI decreases when the similarity of two samples belonging to the same cluster increases or the dissimilarity of two samples belonging to different clusters increases. The first intuitive version of this algorithm consists in exploring all possible subsets of features and selecting the one with the lowest value of FFEI. The paper suggests also an ad-hoc neural network (NN) whose objective is to minimize the FFEI through unsupervised learning: each pair of samples in the original feature space is presented at the input layer of the NN and the weights of the NN are updated by using a gradient-descent technique aiming at minimizing the FFEI. The stopping criteria of the learning algorithm are the maximum number of iterations or a minimum value of the FFEI. At the end of the learning phase, the weights of the NN represent the relevance of the individual features in characterizing/discriminating different clusters.

The proposed algorithm allows us for defining a ranking of the features according to their contribution to the clusters “separability”. On the contrary, it does not perform a proper
feature clustering. Moreover, the training phase of the NN severely affects the overall computational complexity.

**F. Feature Similarity Selection Algorithm (FSSA) [18]**

The algorithm proposes a novel unsupervised feature selection technique which exploits feature dependency/similarity to reduce redundancy but does not require searching the feature space. The algorithm starts by clustering the features with a k-means algorithm (the value of k is user-defined) according to the

\[ \lambda_2(x, y) = \frac{1}{2} \left( \text{var}(x) + \text{var}(y) - \sqrt{\text{var}(x) + \text{var}(y))^2 - 4 \text{var}(x) \text{var}(y)(1 - \rho(x, y)^2)} \right) \]

figure of merit. \( x \) and \( y \) are feature vectors, \( \text{var}(\bullet) \) denotes the variance and \( \rho(x, y) \) the correlation between \( x \) and \( y \). \( \lambda_2 \), which is the **maximal information compression index**, is the eigenvalue along the direction normal to the principal component and represents the amount of reconstruction error introduced when the dataset is projected to a reduced space in the best possible way [19]. In other words, \( \lambda_2 \) measures the minimum information loss caused by a reduction in the feature number of the dataset. After the clustering phase, the algorithm selects a single feature from each cluster (the other features of the clusters are discarded). The selected features represent the final feature subset.

The proposed algorithm has a very low computational complexity. Moreover, the suggested maximal information compression index is invariant both to the rotation and to the translation of the dataset. On the contrary, it is very sensitive to the scaling transformation as well as by the parameter’s choice.

**Comments:**

As stated in [18], unsupervised feature selection techniques can be subdivided into two main approaches: clustering and ranking. Clustering techniques (i.e., ELSA [15], NSGA [16], FSSA [18]) aim at maximizing clustering performance according to one (or more) figure of merit. On the contrary, ranking methods (i.e., SFFSA [10], SFS [11], NNFS [17]) aim at selecting a subset of features according to their relevance (and removing redundant or irrelevant features). Both approaches can be considered in WSNs for defining the mutual affinity. Clustering techniques provide an explicit relationship among node measurements, i.e., two nodes whose measurements lie in the same cluster have a high affinity in the acquired data. On the other hand, ranking methods provide an implicit relationship among nodes’ measurements, i.e., selected features are the most relevant ones, while discarded features are either irrelevant or highly correlated to one (or more) selected features.

The choice of the final approach depends on the WSN designer’s needs. In principle, clustering techniques could be advantageous for routing end energy management while ranking methods may be very useful for distributed decision algorithms. Given that the approach (clustering or ranking) may be a designer’s choice, the computational complexity of the algorithms might be a strong drawback which only rarely can be neglected. With this in mind, we suggest not to consider ELSA, NSGA, NNFS in WSNs for their excessive computational complexity but opt for SFFSA, SFS and FSSA.

Among these low-complexity algorithms, SFFSA is the only one which relies on strong theoretical bases. Unfortunately, its semi-supervised nature (it requires a reference model in the feature selection phase) makes it less effective. SFS is fast and effective but requires at least 10 features to work properly. Finally, FSSA has a low computational complexity and is not affected by rotations and translations of the dataset but requires a difficult configuration phase.

As stated above, none of the low-complexity algorithm clearly emerges from the others. A wide experimental campaign, presented in the next section, has been performed to experimentally evaluate and compare their feature selection capabilities on some benchmarks.

III. EXPERIMENTAL RESULTS

To compare the performance of the algorithms presented in the previous section we considered six benchmarks from UCI Machine Learning Repository [20]. More in detail,

- **App. - A1** refers to the iris plant dataset (3 classes, 150 samples, 4 features);
- **App. - A2** refers to the breast cancer Wisconsin diagnostic dataset (2 classes, 684 samples and 9 features);
- **App. - A3** refers to the classification of radar returns from the ionosphere (2 classes, 351 samples, 34 features);
- **App. - A4** refers to the prediction of the cellular localization sites of proteins (10 classes, 1484 samples, 8 features);
- **App. - A5** refers to GIS data representing the forest cover type of a region (8 classes, 581012 samples, 10 features);
- **App. - A6** refers to a speech recognition application (26 classes, 7797 samples, 617 features).

We compared the presented algorithms according to the following figures of merit:

- **Ratio r** between the dimension of the feature vector in the selected and the original space;
- **Computational Time (CT)** defined as the execution time (in seconds) needed to execute the unsupervised feature selection algorithm (reference platform: Intel Centrino 1.7 GHz, 1Gb RAM, Win XP, unnecessary processes aborted);
- **Entropy E** measuring the quality of clusters (see [21]); low values of \( E \) imply well-separated clusters;
- **Fuzzy Feature Evaluation Index (FFEI)** defined in [17] and presented in Section II.E;
- **Representation Entropy (RE)** measuring the information compression made possible by dimensionality reduction [19]. High values of \( RE \) imply minimum redundancy in the selected feature subset;
- **Ratio a** between the k-NN classification accuracy by using only the selected feature subset and the original feature set.

The accuracy is evaluated with 10-fold cross-validation. The
The ability to reduce the dimension of the original feature vector is well-explained by the first column of Table 1 (and Table 3): \( r \), which is the ratio between the dimension of the feature vector in the selected and the original space, is generally lower than 0.5. This means that unsupervised feature selection algorithms are generally able to halve the size of the feature vector. SFS is particularly promising since it achieves, on the average, \( r = 0.4 \), while NSGA provides the minimum reduction ability. NNFS achieves the minimum value of \( r \) but to the detriment of the knowledge about the physical phenomenon under monitoring (see the low value of \( a \); this issue will be addressed in more detail below); this algorithm was not able to correctly identify the best trade-off between reduction and loss in information.

Results about the computational time (second column of Table 1 and Table 3): \( CT \), which is the time required for the whole selection process on a single available CPU core, are of particular interest for WSNs. All algorithms require high computational complexity. Table 3 provides very interesting and detailed results about computational complexity: NNFS does not terminate with dataset A5 (more than 580000 samples) and datasets A3, A6 (more than 30 features). Moreover, the two genetic-based algorithms ELSA and NSGA did not terminate with dataset A5. On the contrary, SFFSA, SFS and FSSA have always completed their computation and provide satisfactory computational times. FSSA has a \( CT \) higher than the others but, as presented in Table 3, this is mainly due to A5. By considering the other datasets, FSSA has a computational complexity that is generally ten times greater than SFFSA and SFS.

Low values of \( E \) imply well-formed clusters. Table 1 presents similar values of \( E \) for all the algorithms (apart from NNFS) suggesting similar clustering abilities. NNFS deserves additional comments: the low value of both \( r \) and \( E \) show the characteristic of this algorithm to privilege single features in the final feature subset. Obviously, this approach allows both to reduce the number of considered features and provide well-formed clusters at the cost of a high loss in information content (low values of \( a \)).

\( FEEI \), which provides information about the intra/inter cluster similarity is particularly low for SFSSA and SFS. This implies a particular ability to cluster similar features and dislocate dissimilar features in different clusters.

SFSSA and SFS provide the highest values of \( RE \) and, then, a minimum redundancy in the selected feature set. ELSA and NSGA share a similar behavior, while FSSA provides the lowest performance.

All algorithms (apart from NNFS) provide values of \( a \) which are greater than 0.9 meaning that the loss in accuracy introduced by the feature selection is less than 10\%. In particular, FSSA provides the highest value of \( a \) guaranteeing that, in the considered datasets, the reduction of the feature set does not influence the classification accuracy. This implies that FSSA correctly removed irrelevant or redundant features. On the contrary, at stated before, NNFS discarded relevant features. This is evident by considering the low value of \( a \): the feature reduction has non-negligible negative effects on the classification accuracy.

**Comments**

The experimental campaign corroborated the theoretical comments of Section II. As expected, ELSA, NSGA and NNFS suffer from an excessive computational complexity that makes them, as they are, unfeasible choices for WSNs. Among the other algorithms, FSSA results to have computational times 10 times larger than SFSSA and SFS. Apart from NNFS, which suffers from the “single-feature selection” problem (as shown in the experimental phase), SFS has the highest value of \( r \) guaranteeing good reduction capabilities. On the contrary, as presented in Table 3, SFS has lower performance in case of datasets with less than 10 features. SFSSA guarantees reduction capabilities and computational times slightly greater than SFS but provides well-formed clusters (i.e., lowest values of \( E \)) and minimum redundancy in the selected feature subset (i.e., highest value of \( RE \)).

As a final comment, no winning unsupervised feature selection algorithm clearly emerges from both the theoretical and the experimental analysis. The authors suggest to consider SFFSA for its strong theoretical basis.

**IV. UNSUPERVISED FEATURE SELECTION ON WSN’S NODES**

To address the applicability of SFFSA in WSNs we experimentally evaluated its computational complexity in a real WSN node. The node we considered is the Crossbow Stargate board [22], which integrates a 32-bit 400 MHz Intel PXA255 XScale RISC processor, 32 MB flash memory and 64 MB of SDRAM providing also PCMCIA, Ethernet, USB and serial connectors. This board is particularly appealing since the Linux operating system runs onboard. Such availability allows the designer for simplifying code prototyping and testing. Moreover, such board has reduced sizes (3.5” x 2.5”) and low power consumption (< 500 mA).
Generally, in WSNs, the Stargate board is meant to act as a gateway for a cluster of low level sensors, e.g. motes (see Figure 2). This approach well suits our case since the SFFSA algorithm, executed at the cluster heads, allows for exploiting the mutual affinity inside each cluster by identifying those nodes which provide meaningful information about the physical phenomenon under monitoring (and, conversely, neglecting those nodes providing irrelevant or redundant data).

![Figure 2 - A WSN showing a hierarchical organization of units.](image)

Table 2 provides the computational time (s) averaged over 100 simulations of the SFFSA algorithm executed on the Stargate board with different WSN configurations; the number of samples ranges from 100 to 680 samples. In particular, as far as the WSN configuration is concerned, we simulated clusters composed of 4 nodes (3 low-level nodes and the gateway), 7 nodes (6 low-level nodes and the gateway) and 10 nodes (9 low-level nodes and the gateway). For these simulations we considered data randomly extracted from dataset A2.

Simulation results show that the computational time scales approximately linearly w.r.t. the number of samples and more linearly w.r.t. the number of nodes in the cluster.

Table 2 - Computational times of the SFFSA algorithm executed on the Stargated board

<table>
<thead>
<tr>
<th>Number of samples</th>
<th>Number of nodes in the cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>0.29</td>
</tr>
<tr>
<td>200</td>
<td>0.51</td>
</tr>
<tr>
<td>400</td>
<td>0.99</td>
</tr>
<tr>
<td>680</td>
<td>1.66</td>
</tr>
</tbody>
</table>

In case of small clusters (e.g. 4 nodes), the computational time remains lower than 1s up to 400 samples. On the contrary, larger clusters and a higher number of nodes might require computational times up to 9s. Nevertheless, it is particularly important to emphasize that SFFSA, in its current version [10], requires the whole dataset to operate while an iterative version of the algorithm, under study by the authors, should heavily reduce the computational burden by considering one sample at the time instead of the whole dataset.

V. CONCLUSIONS

The paper presents six most relevant unsupervised feature selection algorithms suggested in the literature and evaluates their usability in Wireless Sensor Networks. The computational complexity has been a strict constraint: only SFFSA, SFS and FSSA algorithms could be considered in WSNs for their reduced computational complexity. However, no winning candidate clearly emerges and the final choice is application dependent. As a final remark, authors would slightly suggest SFFSA, for its strong theoretical justification.

REFERENCES

Table 3 - Simulation Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm</th>
<th>$r$</th>
<th>$CT$</th>
<th>$E$</th>
<th>$FEEI$</th>
<th>RE</th>
<th>$a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 (Iris)</td>
<td>SFFSA</td>
<td>0.500</td>
<td>0.002</td>
<td>0.462</td>
<td>0.347</td>
<td>0.96</td>
<td>0.570</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>0.250</td>
<td>0.002</td>
<td>0.403</td>
<td>0.343</td>
<td>0.72</td>
<td>0.467</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>0.500</td>
<td>0.005</td>
<td>0.455</td>
<td>0.352</td>
<td>0.82</td>
<td>0.566</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>0.750</td>
<td>1.494</td>
<td>0.395</td>
<td>0.346</td>
<td>1.01</td>
<td>0.573</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>0.250</td>
<td>0.083</td>
<td>0.466</td>
<td>0.346</td>
<td>0.56</td>
<td>0.421</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.500</td>
<td>0.063</td>
<td>0.498</td>
<td>0.347</td>
<td>1.01</td>
<td>0.521</td>
</tr>
<tr>
<td>A2 (Cancer)</td>
<td>SFFSA</td>
<td>0.556</td>
<td>0.005</td>
<td>0.234</td>
<td>0.351</td>
<td>0.99</td>
<td>0.936</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>1.000</td>
<td>0.006</td>
<td>0.234</td>
<td>0.351</td>
<td>0.89</td>
<td>0.936</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>0.444</td>
<td>0.040</td>
<td>0.238</td>
<td>0.360</td>
<td>0.99</td>
<td>0.885</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>0.556</td>
<td>4.517</td>
<td>0.240</td>
<td>0.359</td>
<td>0.99</td>
<td>0.908</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>0.110</td>
<td>50.058</td>
<td>0.141</td>
<td>0.363</td>
<td>0.81</td>
<td>0.720</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.444</td>
<td>0.030</td>
<td>0.240</td>
<td>0.367</td>
<td>0.99</td>
<td>0.915</td>
</tr>
<tr>
<td>A3 (Ionosphere)</td>
<td>SFFSA</td>
<td>0.618</td>
<td>0.034</td>
<td>0.573</td>
<td>0.403</td>
<td>1.04</td>
<td>0.637</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>0.618</td>
<td>0.065</td>
<td>0.565</td>
<td>0.402</td>
<td>1.15</td>
<td>0.652</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>0.324</td>
<td>0.737</td>
<td>0.569</td>
<td>0.385</td>
<td>0.99</td>
<td>0.648</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>0.500</td>
<td>20.507</td>
<td>0.566</td>
<td>0.397</td>
<td>0.91</td>
<td>0.646</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>n.a.</td>
<td>1e6 (*)</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.650</td>
<td>0.335</td>
<td>0.570</td>
<td>0.403</td>
<td>2.489</td>
<td>1.05</td>
</tr>
<tr>
<td>A4 (Yeast)</td>
<td>SFFSA</td>
<td>0.500</td>
<td>0.005</td>
<td>0.615</td>
<td>0.230</td>
<td>1.195</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>0.625</td>
<td>0.008</td>
<td>0.646</td>
<td>0.244</td>
<td>1.343</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>0.375</td>
<td>0.039</td>
<td>0.630</td>
<td>0.228</td>
<td>1.009</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>0.750</td>
<td>5.324</td>
<td>0.599</td>
<td>0.215</td>
<td>1.389</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>0.125</td>
<td>115.238</td>
<td>0.000</td>
<td>0.158</td>
<td>0.000</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.500</td>
<td>0.027</td>
<td>0.612</td>
<td>0.221</td>
<td>1.190</td>
<td>0.96</td>
</tr>
<tr>
<td>A5 (Forest)</td>
<td>SFFSA</td>
<td>0.323</td>
<td>19.792</td>
<td>n.a.</td>
<td>n.a.</td>
<td>3.747</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>0.013</td>
<td>1.622</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.994</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>n.a.</td>
<td>1e6 (*)</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>n.a.</td>
<td>1e6 (*)</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>n.a.</td>
<td>1e6 (*)</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.506</td>
<td>534.940</td>
<td>0.604</td>
<td>0.494</td>
<td>0.454</td>
<td>n.a.</td>
</tr>
<tr>
<td>A6 (Isolet)</td>
<td>SFFSA</td>
<td>0.800</td>
<td>1.797</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.769</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>SFS</td>
<td>1.000</td>
<td>1.637</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.772</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>ELSA</td>
<td>0.500</td>
<td>91.445</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.887</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>NSGA</td>
<td>0.700</td>
<td>870.105</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.699</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>NNFS</td>
<td>n.a.</td>
<td>1e6 (*)</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td></td>
<td>FSSA</td>
<td>0.500</td>
<td>12.433</td>
<td>n.a.</td>
<td>n.a.</td>
<td>0.163</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Legend: (*) the algorithm did not end and was stopped after 1e6 seconds.