Pattern Recognition in Wireless Sensor Networks in Presence of Sensor Failures

JANOS CSIRIK
University of Szeged
Department of Computer Science
Arpad ter 2, Szeged
HUNGARY
csirik@inf.u-szeged.hu

PETER BERTHOLET
University of Bern
Institute of Applied Mathematics
and Informatics
Neubruckstrasse 12, Bern
SWITZERLAND
bertholet@iam.unibe.ch

HORST BUNKE
University of Bern
Institute of Applied Mathematics
and Informatics
Neubruckstrasse 12, Bern
SWITZERLAND
bunke@iam.unibe.ch

Abstract: In the current paper we consider the task of object classification in wireless sensor networks. Assuming that each feature needed for classification is acquired by a sensor, a new approach is proposed that aims at minimizing the number of features used for classification while maintaining a given correct classification rate. In particular, we address the case where a sensor may have a failure before its battery is exhausted. In experiments with data from the UCI repository, the feasibility of this approach is demonstrated.

Key–Words: wireless sensor networks, feature ranking, feature selection, system lifetime, sensor failure

1 Introduction

The problems of feature subset selection and feature ranking have been of paramount importance in the discipline of pattern recognition [1]. It is well known that given features are often noisy, irrelevant, or redundant. Hence, eliminating such features may lead to a better performing pattern recognition system. For a detailed discussion of feature selection we refer to the surveys provided in [1, 5].

In most of the previous papers on feature subset selection, the objective was to find a subset of features that lead to a high performance of the resulting classifier. In the current paper we address the problem of feature selection from a different perspective, namely in the context of wireless sensor networks [6]. The field of wireless sensor networks has become a focus of intensive research in recent years and various theoretical and practical questions have been addressed. One of the most critical issues faced in this domain is the restricted lifetime of the individual sensors, caused by limited battery capacity. Thus keeping the energy consumption of the individual sensors low is a key issue in wireless sensor networks. Assuming that the individual features of a pattern recognition problem to be solved in a wireless sensor network context are provided by the network's sensors, minimizing the energy consumption becomes equivalent to minimizing the number of features to be used.

Various approaches to minimizing energy consumption and maximizing the lifetime of sensors have been proposed. Berman et al. [7] have investigated the efficient energy management in a theoretical model. Krause et al. [8] consider the problem of monitoring spatial phenomena, such as road speed on a highway, using wireless sensors with limited battery life. Wang and Xiao [9] provide a survey on energy-efficient scheduling mechanisms in sensor networks that have different design requirements than those in traditional wireless networks. Sun and Qi [10] present a concept of dynamic target classification in wireless sensor networks. Chatterjea et al. [11] observed that in some applications very large amounts of raw data need to be transported through the wireless sensor network. Tan and Georganas [12] propose a node-scheduling scheme, which can reduce the overall energy consumption of the underlying system, and therefore increase system lifetime, by turning off some redundant nodes. Duarte and Hu [13] classify the type of moving vehicles in a distributed, wireless sensor network.

The current paper is based on the assumption that sensors in a wireless network are only activated upon request from the base station of the underlying system. Each sensor measures one particular feature from the environment and returns its value to the base station, where the classification algorithm is executed.[1]

In this paper we reduce the number of features for the purpose of minimizing energy consumption of the sensors, and thus want to increase the lifetime of the

[1]We suppose that classifier training and validation are executed on the base station as well.

system, while maintaining a certain level of classification accuracy. So we do not necessarily expect that we get a higher recognition performance by reducing the number of features, but aim at extending the lifetime of the classifier.

The rest of this paper is organized as follows. In Section 2, our general approach to feature selection for pattern classification in wireless sensor networks is outlined. Then, a series of experiments are described in Section 3. Finally, in Section 4, we present a summary and discussion, and draw conclusions from this work.

2 General Approach

We assume that a pattern **x** is represented by an N-dimensional feature vector, i.e. $\mathbf{x} = (x_1, \ldots, x_N)$, where x_i is the value of the i-th feature; $i = 1, \ldots, N$. Let $S = \{s_1, \ldots, s_N\}$ be the set of available sensors, where each sensor s_i measures exactly one particular feature $f(s_i) = x_i$ to be used by the classifier. Hence, the maximal set of features possibly available to the classifier is $\{x_1, \ldots, x_N\}$. Furthermore, let $\varphi : S \rightarrow R$ be a function that assigns a utility value $\varphi(x_i)$ to each feature $f(s_i) = x_i$. Let us assume that the utility of a feature x_i is proportional to its ability to discriminate between the different classes an unknown object may belong to.

The basic structure of the algorithm for object classification proposed in this paper is given in Fig. 1. The system uses a base classifier. This base classifier can be a classifier of any type, in principle. For the purpose of simplicity, however, we assume in this paper that the base classifier is a k-nearest neighbor (k-NN) classifier [15].

Having a base classifier at its disposition, the algorithm starts with ranking the sensors in line 1. After this step, the sensors $s_1, \ldots, s_N$ are ordered according to the utility of their features $x_1, \ldots, x_N$, such that $\varphi(x_1) \geq \varphi(x_2) \geq \ldots \geq \varphi(x_N)$. That is, the first sensor yields the most discriminating feature, the second sensor the second most, and so on. Then the algorithm initializes the set F of features to be used by the classifier to the empty set (line 2). Next it iteratively activates one sensor after the other, reads in each sensor's measurement, and adds it to feature set F (lines 3 to 6). Once a new feature has been obtained, statement $classify(F)$ is executed, which means that the base classifier is applied, using feature set F (line 7). Note that a k-NN classifier is particularly suitable for such an incremental mode of operation where new features are iteratively added, because it needs only to compute distances of the unknown object to the training instances, and the distance computations can be

1: rank sensors $s_1, \ldots, s_N$ according to the utility of the their features such that $\varphi(x_1) \geq \varphi(x_2) \geq \ldots \geq \varphi(x_N)$
2: $F = \emptyset$
3: **for** $i = 1$ to N **do**
4:     **if** sensor s_i is available **then**
5:         read feature $f(s_i) = x_i$
6:         $F = F \cup \{x_i\}$
7:         $classify(F)$
8:         **if** $confidence(classify(F)) \geq \theta$ **then**
9:             output result of $classify(F)$ and **terminate**
10:         **end if**
11:     **end if**
12: **end for**
13: output result of $classify(F)$

Figure 1: Algorithm for object classification with limited number of sensor measurements.

performed in an incremental fashion, processing one feature after the other and accumulating the individual features' distances. In line 8, it is checked whether the confidence of the classification result is equal to or larger than a threshold θ. If this is the case the classification result is considered final. It is output and the algorithm terminates (line 9). Otherwise, if the confidence is below the given threshold θ, the next sensor is activated.

Obviously, in order to classify an unknown object, the base classifier uses nested subsets of features $\{x_1\}$, $\{x_1, x_2\}, \ldots, \{x_1, x_2, \ldots, x_i\}$ until its confidence in a decision becomes equal to or larger than threshold θ. While running through the for-loop from line 3 to 12, it may happen that a sensor s_i becomes unavailable due to battery exhaustion or some other cause. In this case, sensor s_i will be simply skipped and the algorithm continues with sensor s_{i+1}. In case none of the considered feature subsets leads to a classification result with enough confidence, the classifier outputs, in line 13, the result obtained with the set F of features considered in the last iteration through the for-loop, i.e. for $i = N$.

An important issue in the algorithm of Fig. 1 is how one determines the confidence of the classifier. Many solutions to this problem can be found in the literature [16]. In the current paper, our base classifier is of the k-NN type.

The algorithm of Fig. 1 takes into account that one or several features may not be available. There are several possible causes for such a case, for example, that a sensor has exhausted its battery or has become faulty for some reason. In this case, the corresponding feature x_i is skipped, and the system continues with sensor s_{i+1}.

In order to rank the features in line 1 of the algorithm, three well-known methods have been used. The first method is Relief [4], which directly yields a ranking of the given features. Secondly, a wrapper approach (WA) in conjunction with k-NN classifiers is applied. The k-NN classifiers use only a single feature each. The features are finally ordered according to their performance on an independent validation set. Thirdly, sequential forward search [3] in conjunction with a k-NN wrapper is applied (WA-SFS). Here nested subsets of features $\{x_{i_1}\}$, $\{x_{i_1}, x_{i_2}\}$,...,$\{x_{i_1}, ..., x_{i_N}\}$ are generated and the ranking is given by the order $x_{i_1}, ..., x_{i_N}$ in which the features are added. For more details of features ranking, we refer to [17].

3 Experiments

The algorithm described in Section 2 was implemented and experimentally evaluated. In the field of wireless sensor networks, there are not many data sets publicly available, especially not for pattern recognition problems. Exceptions are [18], [19]. However, the authors of these papers do not mention any use of the data sets for pattern classification problems. Moreover, no classification benchmarks have been defined for any of these data sets. For this reason, it was decided to use datasets from the UCI Machine Learning Repository [20]. The sensors were simulated by assuming that each feature in any of these datasets is delivered by a sensor. The experiments reported in this paper were conducted on dataset Isolet and Multiple Features (see Table 1). Experiments on other datasets from the UCI repository with similar characteristics gave similar results but are not reported here because of lack of space.

Data	#Inst	#Feature	#Class	Training	Test
Isolet	7797	617	26	6237	1560
MF	2000	649	10	1500	500

Table 1: Datasets and some of their characteristic properties

3.1 First Experiment

The purpose of the first experiment was to study how the value of threshold θ (see line 9 of the algorithm in Fig. 1) influences the classification accuracy and the total number of sensor measurements (i.e. features) used for classification. Instead of computing the total number of sensor readings we measure the life-time of the considered system, i.e. the number of classifications a system is able to perform before the sensors become unavailable, because of battery exhaustion. Hence the aim of the experiment is to measure the lifetime of the system and analyze the trade-off between lifetime and accuracy depending on threshold θ.

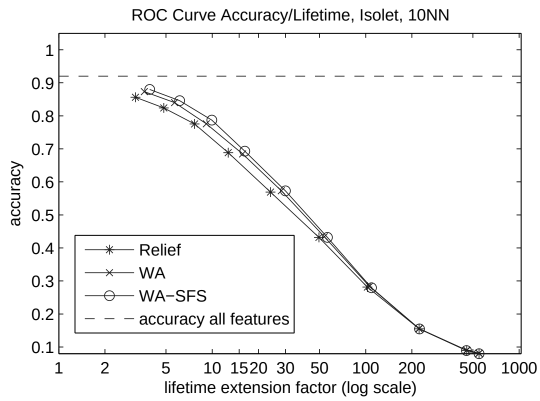

Figure 2: ROC curve of accuracy and lifetime on Isolet

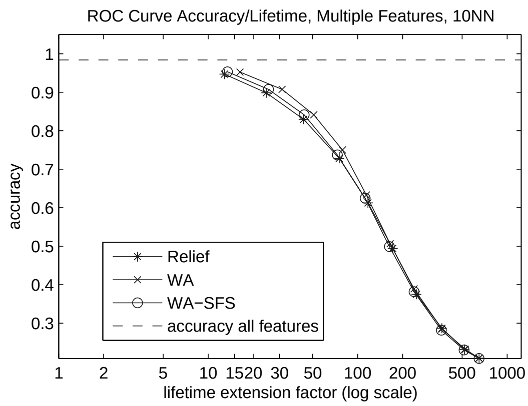

Figure 3: ROC curve of accuracy and lifetime on Multiple Features

We assume that the test set consists of M patterns and each feature x_i can be used exactly M times before the battery of its sensor is exhausted. This means that with a conventional pattern recognition system, which uses the full set of features for each pattern to be classified, the test set can be classified exactly once before all sensors become unavailable. By contrast, with the system proposed in this paper, not all features will be used in each classification step, which allows one to classify the test set multiple times.

In this experiment, we classify the test set multiple times until all sensors become unavailable. Let $M' \geq M$ be the number of pattern instances actually classified, where we count an element of the test set as often as it has been classified. Now we define $lifetime\_ extension\_ factor = M'/M$. Clearly, the $lifetime\_ extension\_ factor$ is bounded by 1 from below. According to our assumption that each feature can be used exactly M times before it becomes unavailable, the case $lifetime\_ extension\_ factor = 1$ occurs if the underlying system always uses all features in each classification step. However, if less than N features are used, the value of the $lifetime\_ extension\_ factor$ will be greater than 1.

In this and the following experiments, we set $k = 10$. We measure the $accuracy$ and the $lifetime\_ extension\_ factor$ both as a function of threshold θ. A representation of the results in terms of ROC curves appears in Figs. 2 and 3. Obviously, we observe a trade-off between $accuracy$ and $lifetime\_ extension\_ factor$. Clearly, in neither of the two datasets the proposed system reaches the accuracy obtained with the full set of features, but for large values of θ (at the left end of the curve) it gets quite close. Without loosing much recognition accuracy, the lifetime of the system can be extended by a factor of more than 3 on Isolet and more than 10 on Multiple Features. For lower values of θ a much higher lifetime extension factor can be achieved, thought at a price of a more pronounced loss of recognition accuracy. Comparing the different ranking strategies we note on Isolet that WA-SFS performs best and Relief worst, while on Multiple Features WA is best and Relief worst. However, the differences are rather small, and one may conclude that the choice of a proper feature ranking strategy is not a critical issue.

3.2 Second Experiment

In real applications, there exist various reasons why a sensor may fail to deliver a measurement upon request from the base station. A failure may be a permanent one, for example, if the battery of the sensor is exhausted, or it may be temporary, for example, if there is some transient distortion in the communication channel between the base station and the sensor. Failures of the first type will be simulated in Experiment 3, while failures of the second type are addressed next.

In the second experiment it was assumed that in line 2 of the basic algorithm, sensor s_i will not be available with a certain probability p. Five different values of parameter $p = 0; 0,01; 0,05; 0,10; 0,50$ were selected. For each of these values, a plot like Figs. 2 and 3 was created. Note that for $p = 0$ plots identical to Figs. 2 and 3 are obtained. In order to make the figures less cluttered, only feature ranking strategy Relief is shown.

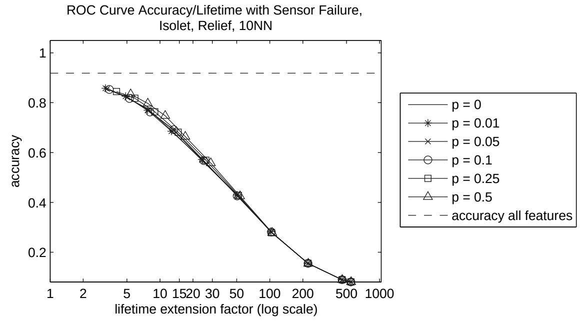

Figure 4: ROC curve of accuracy and lifetime with sensor failure, Isolet

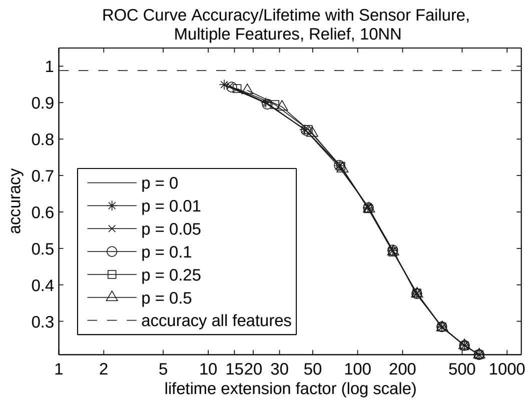

Figure 5: ROC curve of accuracy and lifetime with sensor failure, Multiple Features

The results of this experiment are given in Figs. 4 and 5. Obviously, there is little impact on the bevavior of the system for all tested values of p. From this observation one can conclude that the system is quite stable with respect to occasional sensor faults.

3.3 Third Experiment

From the first experiment one can conclude that the lifetime of a system can be increased at the cost of decreased $accuracy$. However, no quantitative statement can be made about how the decrease in $accuracy$ takes place over time. In the third experiment we proceed similarly to Experiment 1 and classify the test set several times. Yet we do not report the accuracy in the global sense, i.e. in one number for all runs together, but want to see how it changes as the systems evolves over time and more sensors become unavailable.

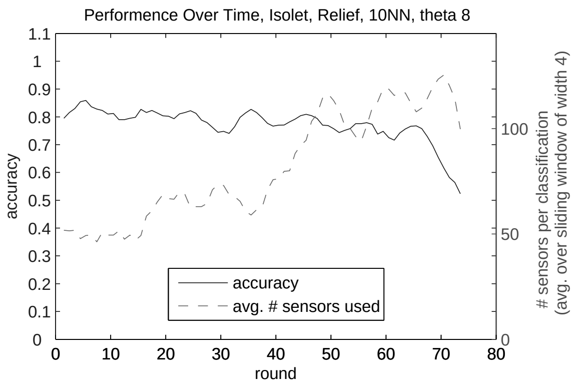

Figure 6: Performance of Relief over Time on Isolet

In Experiment 3 the test set was divided into smaller portions of size one tenth of the original test set size. Then the algorithm of Fig. 1 was applied until all sensors became unavailable. For each portion of the test data the recognition rate and the number of sensors used were recorded. For the sake of brevity, we show only results for threshold $\theta = 8$ and the feature ranking strategy Relief.

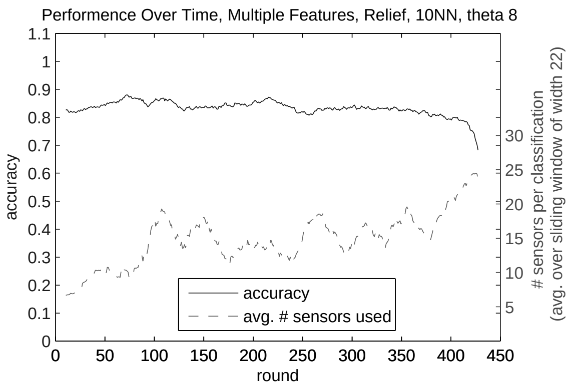

Figure 7: Performance of Relief over Time on Multiple Features

In Figs. 6 and 7, the results of the third experiment are shown. The x-axis depicts the number of rounds through the partitions of the test set, while on the left and right y-axis the $accuracy$ and the number of sensors actually used is given, respectively. On both datasets we observe a similar behavior. On data set Isolet, the accuracy does not decrease much until about round 65. Afterwards it decays very rapidly. The number of features fluctuates remarkably, but shows an upward trend until round 70. Then it quickly declines. The second phase of this decline, when only very few sensors are left, is paralleled by a steep decline of the $accuracy$. From the qualitative point of view, a similar behavior can be observed on data set Multiple Features.

4 Summary, Discussion and Conclusions

In this paper, a sequential multiple classifier system is proposed for reducing the number of features used by a classifier. It is motivated by applications in wireless sensor networks. The procedure can be applied in conjunction with any known method for feature ranking. In the current paper three well known methods, viz. Relief, a wrapper approach based on evaluating each feature individually with a k-nearest neighbor classifier, and a wrapper approach in conjunction with sequential forward search are applied. The underlying base classifier is a k-nearest neighbor classifier.

The proposed procedure was implemented and experimentally tested. As test data, two datasets from the UCI Machine Learning repository were used. A wireless sensor network scenario was simulated by assuming that the individual features are delivered by independent sensors. The results of the experiments revealed that the system behaves very well. Its lifetime can be noticeably increased without loosing much recognition accuracy. During most of its lifetime, the system behaves quite stable. That is, the recognition rate only slightly decreases over the system's lifetime, and a drastic drop happens only towards the very end when almost all sensors are no longer available.

The proposed system can be applied to pattern classification tasks in real wireless sensor networks provided that the objects or events to be classified behave in some stationary way. Because features are acquired in a sequential fashion and the decision of the classifier about the class label of an unknown object or event is only available after the first i features $(1 \leq i \leq N)$ have been processed, it is required that the object or event to be recognized does not change until sensor s_i has delivered feature $f(s_i) = x_i$. This may be a problem when quickly moving objects or rapidly changing events are to be classified. However, there are many potential applications of wireless sensor networks, where this stationary assumption is typically satisfied. Examples include environment monitoring and surveillance.

There are many ways in which the work described in this paper can be extended. First of all one can think of investigating classifiers other than the k-nearest neighbor classifier. Similarly, in addition to the three feature ranking strategies considered in this paper, there are many alternative methods known from the literature [1]. Moreover, an extension of the experiments to more datasets would be desirable, in particular datasets obtained from real wireless sensor networks.

Acknowledgements: This research has been supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

References:

[1] Gyon I, Gunn S, Nikravesh M, Zadeh L (eds.) (2006) Feature Extraction, Foundations and Applications, Springer

[2] Jolliffe I (1986) Principal Component Analysis. Springer

[3] Pudil P, Novovicova J, and Kittler J (1994) Floating search methods in feature selection. Pattern Recognition Letters, 15(11): 1119-1125

[4] Kira K, & Rendell LA (1992) The feature selection problem traditional methods and a new algorithm, In Proceedings of the 9th National Conference on Artificial Intelligence: 129-134

[5] Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. Journal of Machine Learning Research 3: 1157-1182

[6] Xiao Y, Chen H, Li FH (2010) Handbook on Sensor Networks, World Scientific

[7] Berman P, Calinescu G, Shah C, Zelinovsky A (2006) Efficient energy management in sensor networks, In: Ad hoc sensor networks (Eds. Yi Pan, Yang Xiao), Nova Science Publisher

[8] Krause A, Rajagopal R, Gupta A, Guestrin C (2009) Simultaneous placement and scheduling of sensors, IPSN'09, April 13-16, San Francisco, ACM Press: 181-192

[9] Wang L, Xiao Y (2006) A survey of energy-efficient scheduling mechanisms in sensor networks, Mobile Networks and applications, 11: 723-740

[10] Sun Y, Qi H (2008) Dynamic target classification in wireless sensor networks, Proc 19th Int. Conf. on Pattern Recognition

[11] Chatterjea S, Nieberg T, Meratnia N, Havinga P (2008) A distributed and self-organizing scheduling algorithm for energy-efficient data aggregation in wireless sensor networks, ACM Trans. On Sensor Networks, Vol. 4, No. 4, Article 20

[12] Tian D, Georganas ND (2002) A Coverage-preserving node scheduling scheme for large wireless sensor networks, WSNA, ACM Press: 32-41

[13] Duarte MF, Hu YH (2004) Vehicle classification in distributed sensor networks, J. Parallel Distr. Comput. 64: 826-838

[14] Last M, Bunke H, and Kandel A (2002) A feature-based serial approach to classifier combination. Pattern Analysis and Applications, 5: 385-398

[15] Duda RO, Hart PE, Stork DG (2001) Pattern Classification, Second Edition, Wiley-Interscience

[16] Chow CK (1970) On optimum error and reject trade-off. IEEE Trans. on Information Theory, IT-16(1): 41-46

[17] Csirik, J., Bunke, H.: Feature selection and ranking for pattern classification in wireless sensor networks, to appear in P. Wang (ed.): Pattern Recognition, Machine Intelligence and Biometrics Expanding Frontiers. Springer (2011)

[18] Zappi P, Lombriser C, Farelle E, Roggen D, Benini L, Troester G (2009) Experiences with experiments in ambient intelligence environments, IADIS Int. Conference Wireless Applications and Computing

[19] Intille S, Larson K, Tapia EM, Beaudin JS, Kaushik P, Nawyn J, Rockinson R (2006) Using a live-in laboratory for ubiquitous computing research, Proc. PERVASIVE 2006, Springer LNCS 3968: 349-365

[20] Frank A, Asuncion A (2010) UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[21] Mitchell TM (1997) Machine Learning, Mc Graw-Hill

[22] Kuncheva LI (2004) Combining Pattern Classifiers, Wiles