Poster Abstract: Energy-Efficient Distributed Support Vector Machines for Wireless Sensor Networks

K. Flouri
Department of Computer Science,University of Crete
FORTH-ICS,Greece
flouri@ics.forth.gr

B. Beferull-Lozano
Instituto de Robótica
Universidad de Valencia
ETSI, Spain
baltasar.beferull@uv.es

P. Tsakalides
Department of Computer Science,University of Crete
FORTH-ICS,Greece
tsakalid@ics.forth.gr

ABSTRACT

As the research field of mobile computing and communication advances, so does the need for a distributed, ad-hoc wireless network of hundreds to thousands of microsensors, which can be randomly scattered in the area of interest. In this paper, we present two energy-efficient algorithms to perform distributed incremental learning for the training of a Support Vector Machine (SVM) in a wireless sensor network, both for stationary and non-stationary sample data (concept drift). Through analytical studies and simulation experiments, we show that the two proposed algorithms exhibit similar performance to the traditional centralized SVM training methods, while being much more efficient in terms of energy cost.

1. INTRODUCTION

One of the most significant tasks to be performed by a wireless sensor network (WSN) is classification, that is, the action to infer whether the samples measured by the sensors belong to a certain hypothesis or not. It is well known that Support Vector Machines have been successfully used as classification tools in a variety of areas [1]. Training a SVM calls for solving a quadratic programming (QP) problem, in a number of coefficients equal to the number of training examples. Because of this fact, for very large data sets standard numeric techniques for QP become infeasible. In addition, already proposed incremental optimization approaches are not useful for true distributed learning in the context of WSNs, where there exist important constraints in terms of memory and power available at the sensor nodes.

On the other hand, an appealing feature of SVMs that make them well suited to be trained incrementally is the sparseness representation of the decision boundary they provide. The location of the separating hyperplane is specified via real-valued weights on the training samples. But only training samples that lie close to the decision boundary between the two classes, the so-called *support vectors*, receive non-zero weights. In fact, since SVMs can be specified by a small number of support vectors, as compared to the total number of training samples, they provide a compact representation of the data to which new examples can be added as they become available.

In Section 2, we take advantage of this compact representation in order to propose two energy-efficient distributed learning algorithms for WSN deployments. In Section 3, we present a set of simulation experiments in order to assess the performance of our proposed approaches vis-à-vis the performance of a representative centralized SVM algorithm. Finally, we verify the energy efficiency of the new algorithms through analytical studies of the energy cost in both the decentralized and centralized cases.

2. DISTRIBUTED TRAINING OF A SVM

Let us consider a deployment of sensors taking measurements in a certain area. Our goal is to be able to train a SVM in an efficient and distributed fashion so that: a) we can get good classification results on test data and b) our algorithms can be used easily in the context of WSN, where the training must take place across sensors. With this motivation, we propose two novel distributed algorithms in order to train incrementally a SVM in a WSN scenario using an energy-efficient clustering protocol.

A. Distributed Fixed-Partition SVM training: Typical fixed-partition techniques divide the training samples in batches (clusters of sample vectors) of fixed size. This type of algorithms seems appropriate for training incrementally a SVM using *only* partial information at each incremental step [4]. For the WSN scenario, we propose a Distributed Fixed-Partition algorithm (DFP-SVM) where the final estimation of the separating hyperplane is obtained incrementally through a sequence of estimation steps that take place at each data cluster. The key idea behind this incremental algorithm is that instead of transmitting to the next clusterhead all the measurements of the previous cluster, only the current estimates of the hyperplane-defining support vectors are transmitted, thus reducing significantly the power spent for communication (cf. Figure 1).

As we show in our experimental results of Section 3, after only a complete pass through all the clusters, a good approximation of the optimal separating plane is obtained, that is, the separating hyperplane is very similar to the one obtained using a centralized power consuming algorithm, where all the sample data must be transmitted to a central location for processing.

B. Weighted DFP-SVM training: In many real world applications, the concept of interest (definition of classes to be separated) may be time-varying or space-varying; similarly, the underlying data distribution may change as well. Often these changes make the model built on old data inconsistent with the new data, hence regular updating of the model is necessary. This problem, known as *concept drift*, complicates the task of learning in SVM. An example where data distribution changes over space is vehicle tracking for surveillance or monitoring of a hostile environment. In this case, sensors should track all kinds of vehicles that pass through the area and probably have different characteristics such as weight, size, and shape.

In the case of distributed sequential training of a SVM in a WSN, this effect is even more accentuated: As the data is presented in several batches, changes in the target concept may occur between different batches of data. To address this problem, one needs to make the error on the old support vectors (representing the old learning set), more costly than the error on the new samples.

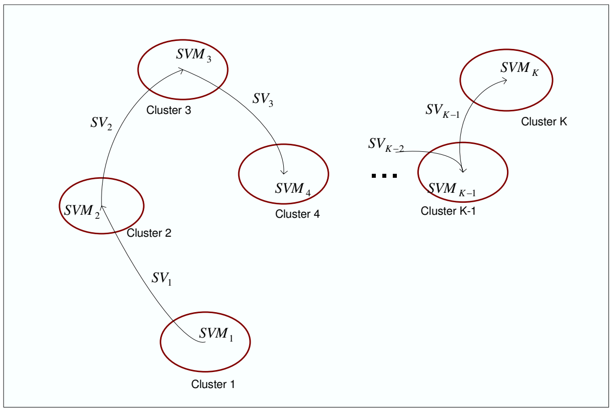

Figure 1: Scheme of distributed training of a SVM: For each cluster, the estimation SVM_i at clusterhead i is obtained combining the support vectors (SV_{i-1}) of the previous estimation SVM_{i-1} calculated at cluster $i-1$ and all the sample vectors measured by the sensors belonging to cluster i.

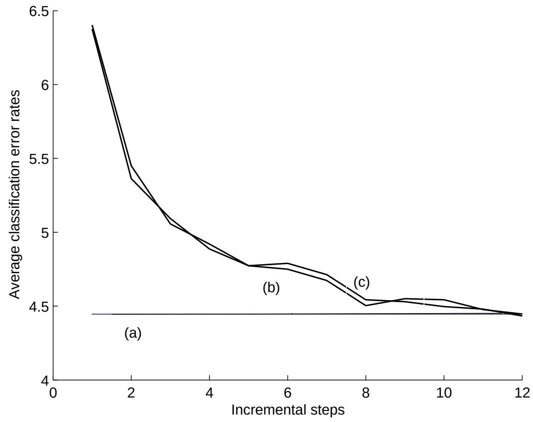

Figure 2: Performance of the training algorithms: The average error rate of 500 Monte Carlo runs after training the SVM for consecutive incremental steps applying the centralized algorithm (line (a)), DFP-SVM (curve (b)) and WDFP-SVM (curve (c)).

Therefore, we modify our previously proposed algorithm according to this observation in order to make it more suitable for WSN applications where there exist concept drifts. Our approach consists of adapting Ruping algorithm [3] to the WSN context. We call this algorithm the Weighted Distributed Fixed-Partition SVM training (WDFP-SVM).

3. RESULTS AND DISCUSSION

In this Section, we evaluate the performance of the two proposed distributed algorithms in terms of the average classification error rate and we compare them to the traditional centralized SVM training algorithm, which requires sensor nodes to forward all the information contained in the observations to a classification center [2]. At the same time, we verify analytically that the energy consumption decreases when the SVM is trained in a distributed fashion.

We consider a sensor network composed of 300 nodes uniformly distributed in the field, where each of the sensors collects sample vectors from two classes. In our experiments, we generate the sample data of the two classes using two Gaussian distributions with two different means. We simulated 500 Monte Carlo runs in order to test the performance of these two distributed algorithms on this set. Figure 2 represents the average error rates (%) for our two proposed algorithms as a function of the consecutive incremental steps. It is shown that with only one pass across the clusters, both distributed algorithms converge to the same average classification error rate obtained with the centralized algorithm that uses all data.

At this point, we would like to investigate the benefits in terms of energy in a wireless sensor network using these distributed algorithms for training a SVM. Specifically, we are interested in the comparison of energy consumed by the proposed distributed algorithms to a scheme where all sensors transmit their data to a fusion center for processing.

Consider the arrangement of n sensors in a cubic lattice where each sensor is at distance d of a neighbor sensor. Now, separate the sensors in K clusters of $(2k+1) \times (2k+1)$ sensors each. Each sensor consumes $E_K(d)$ energy for transmitting its measurements to the clusterhead and each cluster consumes $E_{sv}(d)$ energy for the transmission of N_i support vectors to the next clusterhead $i+1$.

The total energy consumed for the distributed training of a SVM using the proposed algorithms is $E_d(d) = E_{sv}(d) + E_K(d)K$, or:

$$E_d(d) = (2k+1)d^2(N_1 + N_2 + ... + N_{K-1})$$
$$+(6d^2k(k+1) + 8d^2 \sum_{j=1}^{k-1}\sum_{i=1}^{k-j} 2(k-i) + \sum_{j=1}^{k-1} j(k-j))K.$$

On the other hand, the power cost for the direct transmission of the measurements of $(2k+1) \times (2k+1)$ sensors to the base station is given by the expression:

$$E_c(d) = 8d^2 \sum_{j=1}^{k-1}\sum_{i=1}^{k-j}(i^2 + (k-i+1)^2) + 2d^2k(k+1)(2k+1).$$

We simulated 500 Monte Carlo runs in order to estimate the power consumed during the distributed training of a SVM. For a scenario of $n = 225$ sensors in a square grid arrangement separated in $K = 9$ clusters consisting of 25 sensors each (hence $k = 2$), the power cost for the training of the SVM using the proposed distributed algorithm is $E_d(d) = 3380d^2 + 9 \cdot 60d^2 = 3920d^2$, while in the centralized case the cost is $E_c(d) = 8400d^2$. This simulation experiment shows that the proposed distributed algorithm is much more efficient in terms of energy consumption than the centralized algorithm, since it reduces the energy cost by more than 50%.

4. REFERENCES

[1] K. Bennett and C. Campbell. Support vector machines: Hype or hallelujah. *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations*, 2,2:1–13, 2000.

[2] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proc. IEEE Workshop on Neural Networks and Signal Processing, NNSP'97*, pages 276–285, Amelia Island, FL, September 1997.

[3] S. Ruping. Incremental learning with support vector machines. In *Proc. IEEE Int. Conf. on Data Mining*, pages 641–642, San Jose, CA, USA, November 2001.

[4] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence, IJCAI'99*, July 1999.