16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

DISTRIBUTED CONSENSUS ALGORITHMS FOR SVM TRAINING IN WIRELESS SENSOR NETWORKS

K. Flouri[1]*, B. Beferull-Lozano*[2]*, and P. Tsakalides*[1]

[1]Department of Computer Science
University of Crete and
Institute of Computer Science (FORTH-ICS)
71110 Heraklion, Crete, Greece
{flouri,tsakalid}@ics.forth.gr

[2] Instituto de Robótica
Escuela Técnica Superior de Ingeniería (ETSI)
Universidad de Valencia (UV)
46071, Valencia, Spain
Baltasar.Beferull@uv.es

ABSTRACT

This paper studies coordination and consensus mechanisms for Wireless sensor networks in order to train a Support Vector Machine (SVM) classifier in a distributed fashion. We propose two selective gossip algorithms, which take advantage of the sparse representation that SVMs provide for their decision boundary (hyperplane), in order to ensure convergence to an optimal or close-to-optimal classifier, while minimizing the required amount of information exchange between neighboring sensors. The first proposed algorithm calls for the local exchange of support vectors between sensors, while the second technique requires the exchange of all sample vectors that define uniquely and completely the convex hulls of the two classes. Through simulation experiments, we show that the proposed algorithms achieve a consensus close to the desired hyperplane obtained with a centralized SVM-based classifier that uses the entire sensor data.

1. INTRODUCTION

As the research field of mobile computing and communication advances, so does the idea and the need of deploying a distributed, ad-hoc wireless network of hundreds to thousands of microsensors, which can be randomly scattered in the area of interest. Wireless sensor networks (WSNs) enable a variety of new applications such as environmental monitoring, warehouse inventory tracking, location sensing, patient and structural health monitoring. Moreover, in the near future, the development of visual sensor networking technology employing content-rich vision-based sensors will require efficient distributed processing for automated event detection and classification.

The fact that data are collected by sensors at geographically distinct locations necessitates the design of intersensor communication and local information processing, while keeping energy consumption low. One of the most important tasks to be performed in a WSN, is classification, that is, it is important to infer whether the samples measured by sensors in a WSN belong to a certain hypothesis (class) or not. It is well known that Support Vector Machines have been successfully used as classification tools in a variety of areas [1, 2, 3]. Various incremental algorithms have been recently proposed [4, 5, 6, 7] for training a SVM. The key idea in all of them is to preserve only the current estimation of the decision boundary at each incremental step along with the next batch of data (or part of it).

A disadvantage of these techniques is that they may give only an approximate solution and may require many passes through the whole data set to reach a reasonable level of convergence. In principle, all working methods used to train SVMs, especially shrinking [8], use only a small part of the samples for optimization in each step. This is because in all these methods, none of the samples are discarded during the training and thus all of them have to be considered in each working set selection step. As a consequence, both the memory and the power required are too high to be used in WSNs.

In our previous work [9, 10], we proposed two energy-efficient algorithms that involve a distributed incremental learning for the training of a SVM in a WSN. In all incremental techniques, the update of the estimate is diffused sequentially in the network and the convergence to the global estimate is reached at the final step of the algorithm. Hence, at each time slot only one node has the updated critical information and consequently the optimal estimate. In this case, the trained SVM classifier is constructed at the final step of the algorithm. However, nodes in a WSN, usually operate in environments that are prone to link and node failure. Hence, it is important to design algorithms that are robust to unexpected failures of nodes and consequently to changes in the topology. Thus, to maximize robustness, all nodes should ideally achieve convergence to the same optimal estimate.

Distributed consensus is broadly understood as agents (sensors) achieving a consistent view of the state of nature by interchanging information regarding their current state with their neighbors. Motivated by applications to sensor networks, gossip algorithms have been studied, for computation and information exchange in an arbitrarily connected network of nodes. Exhaustive research has been made mostly on the averaging problem, where each sensor updates its local estimate by appropriate weighting the estimates of its neighbors [11, 12]. Gossip algorithms are typically based on iterative schemes, whose energy consumption is proportional to the time necessary to achieve consensus and hence the topology of the network [13].

In this paper, we use an inherent characteristic specific to SVMs to propose two distributed consensus algorithms for the efficient training of SVM classifiers in WSNs. Namely, we use the property that the decision hyperplane of a SVM is completely specified by a small fraction of the whole data vectors, the so-called support vectors. In the first scheme, each sensor updates its hyperplane at every iteration by combining its support vectors with the support vectors commu-

This work was supported by GSRT under program ΠΕΝΕΔ, Code 03ΕΔ69 and by the Marie Curie TOK-DEV "ASPIRE" grant (MTKD-CT-2005-029791) within the 6th European Community Framework Program .

nicated by the neighbors. This results in a close-to-optimal efficient distributed scheme. In a second approach, the information exchanged between sensors describes uniquely and completely the convex hulls of the two classes. The paper is organized as follows. In Section 2, we provide a brief description of SVMs. Section 3 presents the two proposed selective gossip algorithms. Finally in Section 4, we illustrate a set of simulation experiments in order to assess the performance of our proposed approaches.

2. SUPPORT VECTOR MACHINES

Given a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, support vector learning tries to find a hyperplane, determined by a vector $\mathbf{w}$ with minimal norm and an offset vector b, that separates the training data $\{\mathbf{x}_i\}$ into two classes denoted by $y_i = \{-1, +1\}$. Let $SVM = \{\mathbf{w}, b\}$ denote the separating hyperplane. To find such a hyperplane, one must solve the following quadratic problem [14]:

$$\min_{\mathbf{w},\xi} \Phi(\mathbf{w}, \xi) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i, \qquad (1)$$

subject to

$$\begin{aligned} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) &\geq 1 - \xi_i \quad \text{and} \\ \xi_i &\geq 0, \quad \text{for } i = 1, 2, \ldots n \end{aligned}$$

where b determines the offset of the plane from the origin, the set of variables $\{\xi_i\}_{i=1}^{n}$ measures the amount of violation of the constraints, and C is a parameter that determines the cost of constraint violation. The vector of minimal norm $\mathbf{w}$ representing the resulting separating hyperplane is given by:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \qquad (2)$$

which is expressed by means of a linear combination of the so-called *support vectors*, *i.e.*, the training sample vectors $\{\mathbf{x}_i\}_{i=1}^{l}$ corresponding to the l *non-zero* Lagrange multipliers $\{\alpha_i\}_{i=1}^{l}$, calculated during the optimization process[1]. In practical settings, the number of support vectors is usually quite small compared to the number of training samples ($l << n$). The decision function for classifying a new point $\mathbf{x}$ can be easily written as

$$f(\mathbf{x}) = sign\,(\sum_{i=1}^{l} y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b) \qquad (3)$$

and the corresponding decision rule can be expressed as follows: a new test vector $\mathbf{x}$ belongs to class 1 when $f(\mathbf{x}) > 0$ while $\mathbf{x}$ belongs to class -1 when $f(\mathbf{x}) < 0$.

Geometrically, the discriminant can be found by exploiting the convex hull of the training data corresponding to each class. The convex hull of a set of points is the smallest convex set containing the points. More specifically, since there are many planes that separate two classes, the best is considered to be the "furthest" from both classes [14]. One can examine the convex hull of each class' training data and then find the closest points in the two convex hulls. In Figure 1, the measurements are depicted in two dimensions and the closest points in the convex hulls are the circles labelled as 'd' and 'c'. The classifier is the plane that bisects these two points (w=d-c). In Figure 2, support vectors are depicted with the circled dots; only these three vectors correspond to the only non-zero Lagrange multipliers resulting from the solution of the optimization problem 1 and therefore are sufficient for the construction of the discriminant in (2).

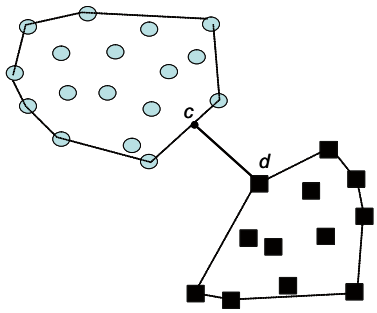


Figure 1: Best plane bisects closest points in the convex hulls.

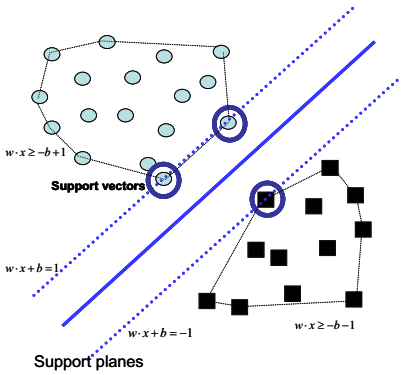


Figure 2: Support Vectors (three circled dots) are sufficient to construct the best plane that separates the two classes.

3. SELECTIVE GOSSIPING FOR SVM TRAINING

Let us consider a deployment of n 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, b) all sensors keep refining their estimate concurrently at each time slot in order to reach finally convergence (consensus) to a common global estimate.

In this work, we use gossip algorithms in the context of a SVM. There is a successive refinement to the estimate of each sensor based on communicating information with one-hop neighbors only. Therefore, at each time slot, the new estimate is diffused to the next-hop neighbors and finally at some point all sensors will reach a consensus. Hence, all sensors in the network converge to the same trained SVM classifier, and can classify any new measurements.

The question at this point is what kind of data should neighboring sensors exchange in order to get high classification accuracy but with low energy consumption? WSN nodes should exchange a sufficient amount of data in order to ensure or approximate optimality. On the other hand, the more data is exchanged, the more energy is consumed. The trade-off between optimality and energy consumption led our research to two different algorithms: a) the Minimum Selective Gossip algorithm (MSG-SVM) where the minimum amount of data is selected for diffusion and b) the Sufficient Selective Gossip algorithm (SSG-SVM) where sufficient data is

[1] Notice that for simplicity, we assume that the sample vectors are enumerated such that the support vectors correspond (in any pre-agreed order) to the first l sample vectors.

Algorithm MSG-SVM

Initialize $\{SV_i(0)\}_{i=1}^{n}$ by training *SVM*
for all sensors with *k* initial measurements

For *t=0,1,2,...* **do**
All *i=1,2,…,n* construct $w_i(t)$
All *i=1,2,…,n* transmit the $SV_i(t)$ to neighbors N_i, Eq.(5)
All N_i update their data
All *i=1,2,…,n* update $w_i(t+1)$ with current data
End for

Figure 3: MSG-SVM algorithm.

diffused to achieve optimality, that is, same performance as a global centralized algorithm. The proposed algorithms are analyzed in Sections 3.1 and 3.2, respectively.

3.1 Minimum Selective Gossip Algorithm (MSG-SVM)

Communication links in the WSN comprised of n sensors, are represented by a graph whose vertices are the sensors and whose edges are formed by the available communication links. The set of sensors having an active link with the i-th sensor are denoted as the neighborhood N_i. The WSN is deployed to train the SVM using the distributed measurements $M_i(0) := \{\mathbf{x}_{i,j}(0)\}_{i=1}^{n}$, where $1 \leq j \leq k$ and k is the total number of measurements acquired by sensor node i.

We begin by taking k measurements at each node i and then training the SVM locally (for each sensor). The first estimate of the hyperplane is denoted by $\mathbf{w}_i(0)$, $i = 1,\ldots,n$, for each node i. When training a SVM, only the support vectors determine the discriminant that separates the data collected by each sensor in two classes [14]. Therefore, the data of each node can be compressed to their corresponding estimated hyperplane and thus to the associated support vectors:

$$SV_i(0) = \{\mathbf{x}_i(0) : \sum_i \alpha_i y_i \mathbf{x}_i(0) = \mathbf{w}_i(0),$$
$$y_i = class\{1,-1\},\ \alpha_i \neq 0\}. \quad (4)$$

In general, it holds that $|\ SV_i(0)\ | << |\ M_i(0)\ |$, where $|\ SV_i(0)\ |$ and $|\ M_i(0)\ |$ denote the cardinality of $SV_i(0)$ and $M_i(0)$, respectively [9].

Our proposed MSG-SVM algorithm is a gossip-based algorithm, where the support vectors $SV_i(0)$ are communicated between one-hop neighbors. Therefore, for each node i, at time $t+1$, we update its estimate $\mathbf{w}_i(t+1)$ by using all the information available at that moment, namely, the previously estimated set of support vectors $SV_i(t)$ at node i, as well as the union of the sets of support vectors $SV_{N_i}(t)$ that have been previously estimated by the neighbor nodes. Notice that once we decide (at a given step $t+1$) what is the new set of support vectors $SV_i(t+1)$ at a given node i, this determines uniquely the corresponding estimate of the hyperplane $\mathbf{w}_i(t+1)$, so there is a one-to-one mapping. The algorithm is described in Figure 3.

The proposed algorithm seems well suited for the distributed training of a SVM in a WSN. To begin with, MSG-SVM is concurrent for each sensor, so finally all n sensors get the measurements that are characterized as support vectors in the n sub-problems. Hence all sensors converge to the same discriminant constructed by those support vectors. Therefore, WSN nodes converge to the same trained SVM classifier, and can classify any new measurements. Additionally, it is an energy efficient algorithm since in order to reduce the energy consumption, each sensor transmits to its neighbors only the support vectors that have not been transmitted in previous steps.

On the other hand, it can be shown that MSG-SVM provides a sub-optimal discriminant hyperplane, with respect to a global centralized algorithm, while communicating the minimum necessary information at each step. As we already mentioned, the data of a node can be compressed to their corresponding support vectors. But it cannot be guaranteed that a vector $\mathbf{x}$ such that $\mathbf{x} \in M_i(t)$ and $\mathbf{x} \notin SV_i(t)$, is not a support vector in $M_i(t+1) = \{M_i(t) \bigcup \cup_{j \in N_i} M_j(t)\}$. In other words, at each step, the set of support vectors associated with the entire data set is not always the same as the overall union of the support vectors obtained after training separately each of the two sets.

Lemma: The MSG-SVM algorithm is sub-optimal, that is, the consensus achieved by training the SVM using only the support vectors from each of the sub-problems is sub-optimal.

Proof: We only need to find a case where a support vector in the training set is not a support vector in any sub-problem. Consider the case of a network comprised of only two sensors collecting measurements in two dimensions. Sensor 1 collects a set of measurements S_1 and the other one collects a set of measurements S_2. For the geometrical aspect of this problem, we need to find a vector in the union of the measurements of both sensors $S = S_1 \cup S_2$ that is a support vector in S, but neither in set S_1 nor in set S_2. Let the smallest distance between the two convex hulls of set S_1 be d_1 and the corresponding distance of set S_2 be d_2, (Figure 4). The convex hull of one class of set S is the smallest set that contains the measurements of set S, hence it also contains the convex hulls of the same class of sets S_1 and S_2. As a counter example one can find a point in the convex hull of set S that is a support vector but it is not a support vector in S_1 nor in S_2. In Figure 5, the squared point is a support vector in S but neither a support vector in S_1 nor in S_2, since the distance between the convex hulls is d, where $d < d_1$ and $d < d_2$.

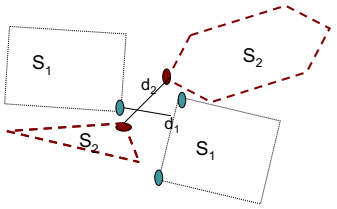


Figure 4: The closest distance between the two dotted convex hulls of S_1 is d_1. Thus, the circled points on the boundary of the convex hulls are the support vectors in set S_1. The closest distance between the two dashed convex hulls of S_2 is d_2. Thus, the circled points on the boundary of the convex hulls are the support vectors in set S_2.

3.2 Sufficient Selective Gossip Algorithm (SSG-SVM)

We propose an alternative algorithm, the Sufficient Selective Gossip Algorithm (SSG-SVM), in order to eliminate the possibility of not converging, such as in MSG-SVM, to the optimal solution. Each sensor sends the amount of data to the

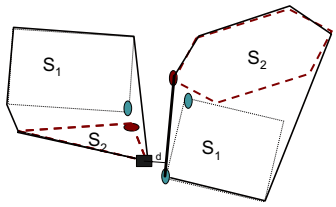


Figure 5: The convex hull depicted in solid line is the convex hull of set $S = S_1 \cup S_2$. The squared point on the convex hull is a support vector in S, since it is one of the closest points between the two convex hulls. Notice that this is not a support vector in S_1 nor in S_2.

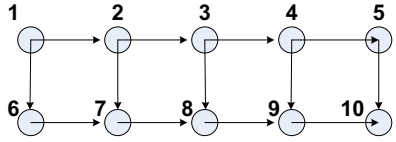


Figure 6: The sensor network is composed of $n = 10$ nodes distributed in a grid topology. The communication links for each sensor are depicted with arrows.

one-hop neighbors that guarantees convergence to the optimal solution. Which is the sufficient amount of data that sensors should exchange to converge to the optimal solution while training a SVM?

We can exploit the geometrical description of a SVM training, illustrated in Figures 1, 2. We examine the convex hull of the training data of each class, and construct the plane that bisects the two closest points of the convex hulls, [14]. This is an alternative equivalent perspective for training a SVM. The closest points can be found by solving the following dual quadratic problem:

$$\min_{\alpha} \frac{1}{2} \parallel \mathbf{c} - \mathbf{d} \parallel^2 \tag{5}$$

$$\mathbf{c} = \sum_{y_i \in class\ 1} \alpha_i x_i, \quad \mathbf{d} = \sum_{y_i \in class\ -1} \alpha_i x_i,$$

subject to

$$\begin{array}{rcl} \sum_{y_i \in class\ 1} \alpha_i & = & 1, \ \sum_{y_i \in class\ -1} \alpha_i = 1 \\ \alpha_i & \geq & 0 \ \text{for } i = 1, 2, \ldots n. \end{array}$$

SSG-SVM takes advantage of the geometrical property of the SVM discriminant hyperplane. The sufficient amount of data for the hyperplane construction are the vectors that lie on the boundary of the convex hulls of the two classes. For each node, the SSG-SVM discards all the vectors of the WSN nodes, except those located at the boundary of the convex hulls. Thus, neighboring sensors exchange the sufficient data only. After some communication, all WSN nodes have the information to construct a separating plane identical to the plane that would have been constructed if all sensors had access to the entire information.

Both algorithms are energy efficient, since data need not be transmitted to a fusion center and the amount of data exchanged by sensors is substantially smaller than the overall generated data. Instead, WSN nodes diffuse partial information to neighboring sensors. Furthermore, each node communicates only information that has not been sent previously, thus the energy spent for transmission is reduced.

4. 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 ideal case where WSN nodes have access to the entire information.

We consider a sensor network composed of $n = 10$ nodes distributed in a grid topology, where each sensor i, $i = 1, \ldots, 10$, collects $\mid M_i(0) \mid = 14$ sample vectors from two classes, at each step. WSN nodes communicate with their one-hop neighbors. The communication links between the sensors in the network are depicted in Figure 6. In our experiments, we generate three sample data sets of two different classes each, using Gaussian distributions with two different means. We choose three data sets, such that their Mahalanobis distance is increasing, Figure 7.

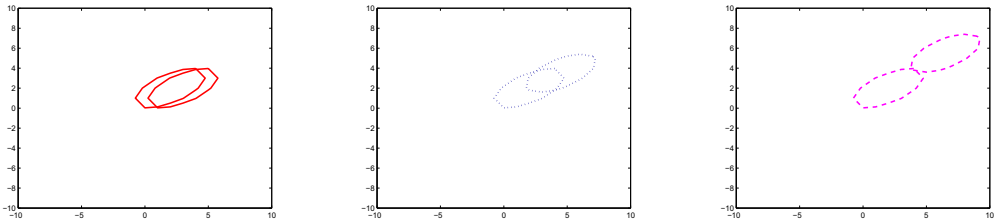
We simulated 100 Monte Carlo runs in order to test the performance of the two proposed Selective Gossip Algorithms. Figure 8 represents the average classification error rates (%) for a randomly chosen sensor, as a function of the iteration steps. After only a few iterations, both algorithms result in trained SVM classifiers which exhibit similar performance to a centralized SVM trained using the entire data from all sensors. SSG-SVM gives an optimal estimate of the discriminant after at most 8 iterations using only partial data. The small divergence of SSG-SVM from the optimal solution (only 2% on average), can be diminished by tuning the parameters of the optimization problem (1). On the other hand, even though MSG-SVM is a sub-optimal solution, it gives a good approximation of the optimal separating plane. Most importantly, with both introduced distributed schemes, all n sensors reach, with a small finite number of steps, an agreement on the nearly optimal discriminant function. Both proposed algorithms behave similarly for all data sets.

The results also show that the difference in performance between MSG-SVM and SSG-SVM is very small. This happens because sensors collect measurements from the same distribution. Therefore, it is very rare to encounter the case where a measurement that is not a support vector in the data set of one sensor, happens to be a support vector in a set containing data from all sensors. In other words, the counter example in Figure 5 is actually an event of low probability; however, such an event may occur more often in scenarios where the class distributions are time-varying.

Moreover, we have also analyzed the trade-off between classification accuracy and energy consumption. Figure 9 illustrates the number of measurements that a particular sensor (tested in data set 2) transmits to its neighbors at each iteration. MSG-SVM gives a sub-optimal solution but uses less measurements than SSG-SVM, thus less energy. SSG-SVM, on the other hand, transmits more data at each iteration, in order to ensure optimality. One can notice that after 5 iterations, in both algorithms, nodes do not need to send any more measurements to their neighbors. After *gossiping* in the network, WSN nodes have exchanged in previous steps all the necessary measurements. Hence, only after a few iterations sufficient amount of data has been diffused to all WSN nodes, each of whom can construct the same trained SVM with the minimum classification error.

5. CONCLUSIONS

In this paper, we propose distributed selective gossip algorithms for training a SVM in a Wireless Sensor Network.



(a) Data set 1 (b) Data set 2 (c) Data set 3

Figure 7: The representation ellipses of different data sets generated by Gaussian distributions.

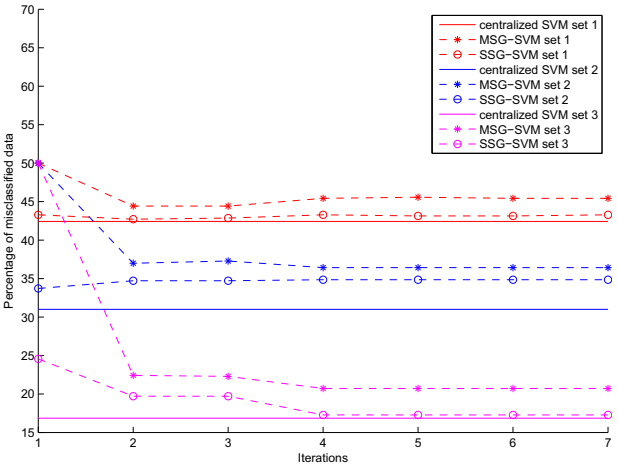


Figure 8: Performance, at a given particular sensor, of the training algorithms for three different data sets. SSG-SVM gives an optimal estimate of the discriminant after at most 4 iterations. MSG-SVM is a suboptimal solution but gives a good approximation of the optimal plane. The ideal case where sensors have access to the entire data is depicted by the straight line.

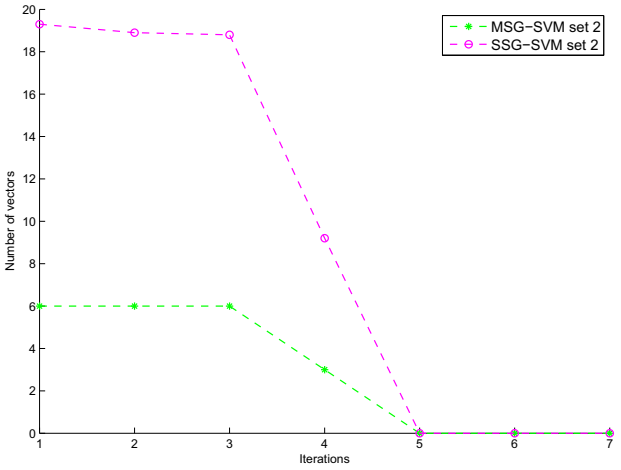


Figure 9: MSG-SVM gives a sub-optimal solution using less measurements than SSG-SVM, which reaches optimality using more data at each iteration.

We introduce two distributed algorithms for training a SVM based on successive refinement of local estimates. In both cases, information is communicated to one-hop neighbors in order to update the estimate at each iteration. The sub-optimal algorithm MSG-SVM, uses only the support vectors of each node to reach an agreement. The SSG-SVM, on the other hand, communicates larger amount of data, *i.e.*, vectors lying on the convex hull boundaries, but converges closer to the optimal solution in a few iterations.

REFERENCES

[1] T. Joachims, "Text categorization with support vector machines," in *Proc. 10th Eur. Conf. on Machine Learning (ECML'98)*, Chemnitz, Germany, April 1998, pp. 137–142.

[2] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. of the 1997 Conference on Comp. Vision and Pattern Recogn.*, Washington, DC, USA, 1997, p. 130, IEEE Computer Society.

[3] A. Bulut, P. Shin, and L. Yan, "Real-time nondestructive structural health monitoring using support vector machines and wavelets," in *Proc. of the Conf. on Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring (NDE'05)*, San Diego, CA, USA, March 2005.

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

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

[6] C. Domeniconi and D. Gunopoulos, "Incremental support vector machine construction," in *IEEE Int. Conf. on Data Mining, ICDM'01*, San Jose, CA, USA, November 2001.

[7] C. P. Diehl and G. Cauwenberghs, "Support vector machine incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. on Neural Networks*, Portland, OR, July 2003.

[8] 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*, Amelia Island, FL, September 1997, pp. 276–285.

[9] K. Flouri, B. Beferull-Lozano, and P. Tsakalides, "Training a SupportVector Machine- based Classifier in Distributed Sensor Networks," in *14nd European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 4-8, 2006.

[10] K. Flouri, B. Beferull-Lozano, and P. Tsakalides, "Energy-Efficient Distributed SupportVector Machines for Wireless Sensor Networks," in *Proc. 2006 European Workshop on Wireless Sensor Networks (EWSN '06)*, Zurich, Switzerland, February 13-15, 2006.

[11] A. Ghosh S. Boyd, B. Prabhakar, and D. Shah, "Gossip Algorithms: Design, Analysis and Applications," in *INFOCOM, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, USA, March 2005, pp. 1653–1664.

[12] A. Ghosh S. Boyd, B. Prabhakar, and D. Shah, "Randomized Gossip Algorithm," *IEEE/ACM Transactions on Networking*, vol. 52,6, pp. 2508–2530, 2006.

[13] Lin Xiao and Stephen Boyd, "Fast Linear Iterations for Distributed Averaging," in *Proc. 42th Conf. on Decision and Control*, Hawaii, USA, December 2003, pp. 4997–5002.

[14] K. Bennett and C. Campbell, "Support vector machines: hype or hallelujah," *SIGKDD Explorations*, vol. 2,2, pp. 1–13, 2000.