Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science (EECSS 2015)
Barcelona, Spain – July 13 - 14, 2015
Paper No. 320

A Two-phase Distributed Training Algorithm for Linear SVM in WSN

Ji Xinrong, Hou Yibin
Beijing University of Technology, Embedded computing Institute
Beijing, China
jixinrong@emails.bjut.edu.cn;yhou@bjut.edu.cn

Hou Cuiqin
Beijing Engineering Research Center for IOT Software and Systems
Beijing, China
houcuiqin@bjut.edu.cn

Abstract- In Wireless Sensor Network (WSN), the centralized methods for training support vector machine (SVM) by all training samples, which are distributed across different nodes and transferred to the fusion centre (FC) by multi-hop routing, will significantly increase the communication overhead and energy consumption. To solve this problem, a novel distributed training algorithm for linear SVM which based on liner kernel function is proposed, which splits the training process of SVM into two phases: the local training phase and the global consensus phase. In the local training phase, aiming at minimizing the difference between each node's local classifier parameters and its local optimal ones which are obtained by exchanging the local classifier parameters with its all neighbours , the quadratic programming (QP) problem of training SVM is derived and solved by using the Lagrange multipliers method. Each node trains its local SVM iteratively until converges. In the global consensus phase, the same classifier parameters can be achieved on each node by using the global consensus algorithm, which relying on the exchanges of final training results on each node between neighbours. Simulation experiments illustrate that the proposed algorithm has better convergence and remarkable advantage in the amount of data transmission comparing with the existing algorithms.

***Keywords*:** Support Vector Machine(SVM); Distributed Learning; Average Consensus; Wireless Sensor Network(WSN)

1. Introduction

As one of the most popular and effective statistical learning method, Support Vector Machine (SVM) has been successfully applied in pattern recognition and classification. In recent years, with the increasing applications of wireless sensor network(WSN), SVM is more and more used in WSN，as shown by Jones et al. (1989), Liu et al. (2013), Aswathy et al. (2012). In WSN, training samples are scattered across different sensor nodes. If each node transfers its local training samples to the central fusion centre (FC) by multi-hop transmission, a more accurate SVM classifier can be obtained by all the training samples in FC. However, this centralized method for training SVM will significantly increase the communication overhead and energy consumption. It contradicts the strictly constraints on bandwidth and energy of WSN and brings the congestion in the nodes around FC. In addition, in some applications which require privacy protection, this training approach is not permitted because of the transmission of training samples. In order to avoid and solve above problems, the distributed training approach for SVM, which only relying on the collaboration between neighbour nodes, is initiated and has attracted more and more interest of the researchers.

In recent years, a lot of research works on distributed learning approach for SVM in WSN have been paid a special attention to and obtain some research achievements. Flouri et al. (2006) and Flouri et al.

(2009) have proposed an incremental algorithm for distributed (D) SVM , its main idea is that the SVs at each clusterhead is obtained by training the SVs passed by the previous clusterhead and its local training samples, and then transmitted to the next clusterhead. After only a complete pass through all the clusters, a separating plane is obtained and as an approximation of the centralized one obtained as if all training samples were centrally available. But this algorithm can't converge to the centralized SVM classifiers because of using only a pass through all the clusters, and if there are a lot of SVs obtained from each cluster, communication overhead will be costly. Forero et al. (2010a,b) have proposed a distributed SVM training algorithm MoM-DSVM based on consensus, which decomposes the centralized linear SVM problem into a set of decentralized convex optimization sub-problems (one per node) with consensus constraints on the classifier parameters of neighbor nodes, and then fully distributed iterative formula are derived using the alternating direction method of multipliers. This distributed algorithm is performed recursively by exchanging their local classifier parameters among the neighbor nodes, so the overhead associated with inter-node communications is fixed and solely dependent on the network topology. However, due to only rely on the collaboration of the neighbor nodes, the convergent speed of the algorithm is low, and its convergent accuracy also needs to be improved. In addition, for solving the training problems of SVM with large training set or distributed data set, many distributed or paralleling algorithms for SVM were proposed as shown by Lu et al. (2008), Wang et al. (2009) and Kim et al. (2012). However, all the nodes can communicate directly to one another and the communication cost and energy consumption in the data transmission has not been considered in these algorithms.

Therefore, this paper presents a novel distributed training approach for linear SVM which is inspired by the idea of the local computation and global communication in distributed algorithms. In this approach, the training of SVM is divided into two phases: the local training phase and the global consensus phase, each phase to use different optimization strategies. In the local training phase, by exchanging its local classifier parameters with its all neighbor nodes, each node trains its local SVM iteratively until converges. In the global consensus phase, consensus can be achieved on each node by using the global consensus algorithm, which relying on the exchanges of final training results on each node between neighbor nodes. The rest of this paper is organized as follows. The proposed distributed algorithm for linear SVM, including the basic ideas, solution technique and derivation, and the realization of algorithm, is described in Section II. Then, the simulation experiments and results are provided in Section III. Finally, Section IV presents the concluding remarks and the future work.

2. Distributed Linear SVM

Given a training set $S=\{(x_i,y_i)\,|\,x_i\in R^d\}_{i=1}^n$, where x_i is an d dimension observation vector, $y_i\in\{-1,1\}$ is the class label of x_i ,and n is the size of S . Training a SVM is to find an optimal separating hyper-plane that separates two classes training examples in S with maximum margin, and this problem can be formulated as a constrained quadratic programming (QP) problem.

In WSN, all training samples are scattered across different nodes and the number of training samples in each node is likely to be different. Therefore, for the linear separable problem, this research adopts the form of quadratic programming problems for linear SVM given in Scholkopf, Smola. (2002), such as shown (1):

$$\begin{aligned}&\min_{w,b,\xi}\ \frac{1}{2}\|w\|^2+\frac{C}{n}\sum_{i=1}^{n}\xi_i\\ &s.t.\ \ y_i(w^T\cdot x_i-b)\geq 1-\xi_i,\forall i\\ &\qquad \xi_i\geq 0\end{aligned}\tag{1}$$

Where $w \in R^d$ is the normal vector to the hyperplane, $b \in R$ is the bias of the hyperplane from the origin, ξ_i are slack variables that permit margin failure , C is a parameter which trades off wide margin with a small number of margin failures and n is the size of training samples.

Consider a WSN with J sensors, any node $j \in J$ only communicates with its one-hop neighbor nodes, and all the neighbor nodes of node j is denoted by $B_j \subseteq J$. Assumed that there is at least one path between any two nodes, i.e., all the nodes in WSN are connected. At each node $j \in J$, only part of the training set is available and is denoted by $S_j := \{(x_{jn}, y_{jn}) : n = 1,..., N_j\}$,where N_j is the number of training samples in S_j. Following the formula (1), only with local training samples, the training of SVM on each node can be expressed as formula (2).

$$\begin{aligned} &\min_{\{w_j, b_j, \xi_{jn}\}} \frac{1}{2}\|w_j\|^2 + \frac{C}{N_j}\sum_{n=1}^{N_j}\xi_{jn} \\ s.t. \quad & y_{jn}(w_j^T x_{jn} + b_j) \ge 1 - \xi_{jn} \qquad \forall j \in J, n = 1,..., N_j \\ & \xi_{jn} \ge 0 \qquad\qquad\qquad\qquad \forall j \in J, n = 1,..., N_j \end{aligned} \tag{2}$$

Where w_j and b_j denote local variables defining a local linear classifier at node j , and other variables have the same meaning as are respectively assigned to them in the formula (1). It should be noted that the value of C in each node is same.

In order to reduce the transmission of a large number of data in training process of SVM, the collaboration between neighbor nodes only by exchanging the local classifier parameters is adopted in this research. A consensus result is obtained in each node by the collaboration between neighbor nodes. This problem may also be viewed as finding an optimal solution in the intersection of each node local constraints. As a result, this problem can be equivalent to looking for the minimum distance between the optimum and the local optimum at each nodeas shown by Bertsekas, Tsitsiklis (1997). Based on this idea, a quadratic term is added to the cost function of (2), the resulting formula is shown as (3):

$$\begin{aligned} &\min_{\{w_j, b_j, \xi_{jn}\}} \frac{1}{2}\|w_j\|^2 + \frac{C}{N_j}\sum_{n=1}^{N_j}\xi_{jn} + \frac{1}{2}\left\|w_j - \overline{w}_j^{\,*}\right\|_2^2 \\ s.t. \quad & y_{jn}(w_j^T x_{jn} + b_j) \ge 1 - \xi_{jn} \qquad \forall j \in J, n = 1,..., N_j \\ & \xi_{jn} \ge 0 \qquad\qquad\qquad\qquad \forall j \in J, n = 1,..., N_j \end{aligned} \tag{3}$$

Where $\overline{w}_j^{\,*}$ is the local optimum of node j , which relies on the intersection of node j 's local constraints and its neighbor nodes' local constraints. Because only the local classifier parameters are exchanged among neighbor nodes, this paper adopts the local consensus algorithm to obtain the local consensus of $\overline{w}_j^{\,*}$, the form is shown as (4). In order to solve the optimization problem (3), the Lagrange multipliers method and dual theory are used. The iterative formulas are derived and shown as (5) and (6).

$$\overline{w}_j^{\,*}(t+1) = w_j(t) - \varepsilon \sum_{i \in B_j}(w_j(t) - w_i(t)) \tag{4}$$

$$\alpha_j(t+1) = \arg \min_{0 \le \alpha_j \le \frac{C}{N_j}} \alpha_j^T A_j A_j^T \alpha_j - [1_j - 2A_j \overline{w}_j^{\,*}(t+1)]^T \alpha_j \tag{5}$$

$$w_j(t+1) = 2*[\overline{w}_j^{\,*}(t+1) + A_j^T \alpha_j(t+1)] \quad (6)$$

In (4), $0 < \varepsilon < 1/d_{\max}$ is a positive scalar parameter, and $d_{\max}$ is the maximum degree of the undirected graph for WSN. In (5), $A_j = \sum_{n=1}^{N_j} y_{jn} x_{jn}$, $\forall j \in J, n = 1,..., N_j$ is the coefficient matrix composed of the training samples. $\alpha_j(t+1)$ is a quadratic programming problem and contains only a single variable, it can be solved by the traditional quadratic programming method. Now the above iterative steps can be executed in parallel on each node, and the local optimum on each node can be achieved through a few iterations.

The resulting SVM training results are very close but not entirely consistent. Hence, in global consensus phase, the consensus algorithm is used to achieve the global consensus on the local training result of each node. In WSN, consensus is one of fundamental tools to design distributed decision algorithm that satisfy a global optimality principle, as corroborated by many works on distributed optimization as shown by Sardellitti et al. (2010). In this research, the simple flooding algorithm for global consensus is adopted. In this algorithm, each node can store the values from its neighbour nodes and forward the received values to its neighbour nodes. All nodes know all values of other nodes in a number of steps equal to the diameter of the graph, then each node can compute the average consensus.

The above training for SVM splits the training process into two phases: the local training phase and the global consensus phase. In the local training phase, by exchanging its local classifier parameters with its all direct neighbor nodes, each node trains its local SVM iteratively until converges. In the global consensus phase, consensus can be achieved on each node by using flooding algorithm for global consensus, which relying on the exchange of final training results on each node between neighbor nodes. Based on this training process, this paper presents a novel two-phase distributed training algorithm for Linear SVM (short as TTL-DSVM), and the overall algorithm proceeds as indicated in TTL-DSVM Algorithm.

TTL-DSVM Algorithm:

Step 1: set k=0, ε equal to a small positive value and initialize w_j [0], $\forall j$

Step 2: Set k=1;

Step 3: Repeat until local convergence

- Compute $\overline{w}_j^{\,*}$ [k], $\forall j$, using (4) ;
- Compute α_j [k], $\forall j$, using (5);
- Compute w_j [k], $\forall j$, using (6);
- Broadcast w_j [k] , $\forall j$, to its neighbors;

k=k+1;

Step 4：Run global consensus on w_j^* and b_j^*, $\forall j$ using flooring algorithm.

3. Numerical Simulations

3. 1. Test case 1: Synthetic Data

Consider a randomly generated WSN with J = 30 nodes and algebraic connectivity 0.0896. The training sets are generated artificially by two different classes random vectors drawn from a 2-dimensional Gaussian distribution with N(2,2,5,5,0) and N(22,2,5,5,0) respectively. Each class of training sets has 600 examples, which are assigned equally to each node of WSN without repeating assignment for examples, thus, each node can acquire 20 examples from each class of training sets. TTL-DSVM has been compared with the centralized SVM(C-SVM) where all the training examples are available, and MoM-DSVM as shown by Forero et al. (2010a,b) in the convergence results and the amount of data transferred.

In Figure 1, two classes of training samples and three decision classification lines obtained by performing C-SVM , TTL-DSVM and MoM-DSVM are shown. Table 1 shows the values of three decision classification lines. As can be seen from three decision classification lines and their values, TTL-DSVM can converge to the results of C-SVM, and MoM-DSVM can obtains the same result of C-SVM in weight vector w , but there is significant different between the results of b of MoM-DSVM and C-SVM.

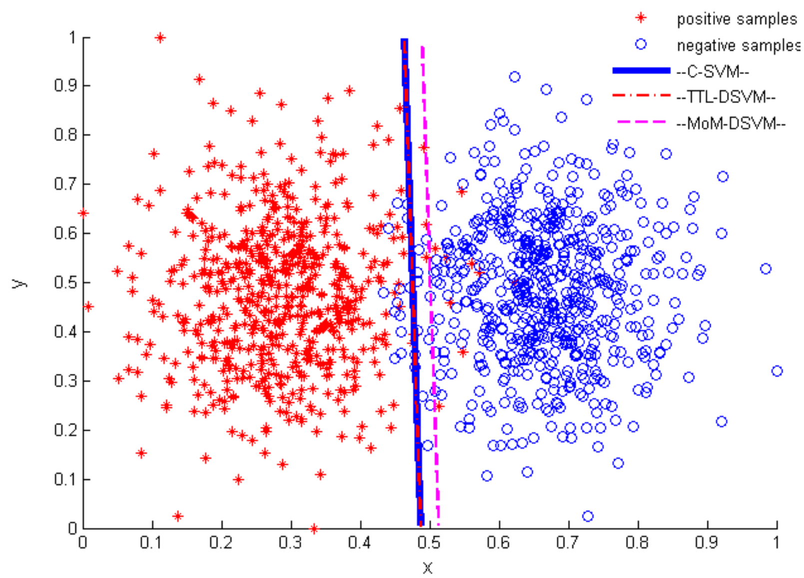

Fig. 1.Two groups of samples and three decision classification lines obtained respectively by C-SVM, TTL-SVM, and MoM-DSVM in Synthetic Data

Table. 1. The w^* and b^* for C-SVM, TTL-DSVM, MoM-DSVM in Synthetic Data

Algorithms	w^* and b^*
C-SVM	$w^* = (-0.3816, -0.0091)^T$; $b^* = 0.1858$
TTL-DSVM	$w^* = (-0.3816, -0.0091)^T$; $b^* = 0.1856$
MoM-DSVM	$w^* = (-0.3816, -0.0091)^T$; $b^* = 0.1956$

Figure 2 shows the amount of data transferred in iteration process of C-SVM, TTL-DSVM and MoM-DSVM. The amount of data transferred of TTL-DSVM is 95.09% less than that of C-SVM, and 43.3% less than that of MoM-DSVM.

3. 2. Test case 2: Wine Quality Dataset

In this case, TTL-DSVM is tested on the Wine Quality Dataset shown by Cortez et al. (2009). Specifically，we use the white wine dataset, which has 11 attributes, 4898 instances and 7 qualities. The binary problem of classifying quality 5 versus quality 7 is considered using a nonlinear classifier. 600 training samples per quality and a test set of 280 samples per quality are taken. Since the obtained classification hyperplane cannot be easily viewed on two-dimensional plane, three components of weight vector and the threshold are chosen to show their iteration processes and convergent results for C-SVM, TTL-DSVM and MoM-DSVM, as shown in Figure 3. The Communication costs for C-SVM, TTL-DSVM and MoM-DSVM in Wine Quality data set are shown in Figure 4, In Figure 3, the three components of w, namely w_1, w_3, w_5, and b for TTL-DSVM have all converged to the results of C-SVM respectively with higher speed. The convergent results of w_1, w_3, w_5 for MoM-DSVM are well close to the results of C-SVM respectively, but there is significant different between the results of b for MoM-DSVM and C-SVM.

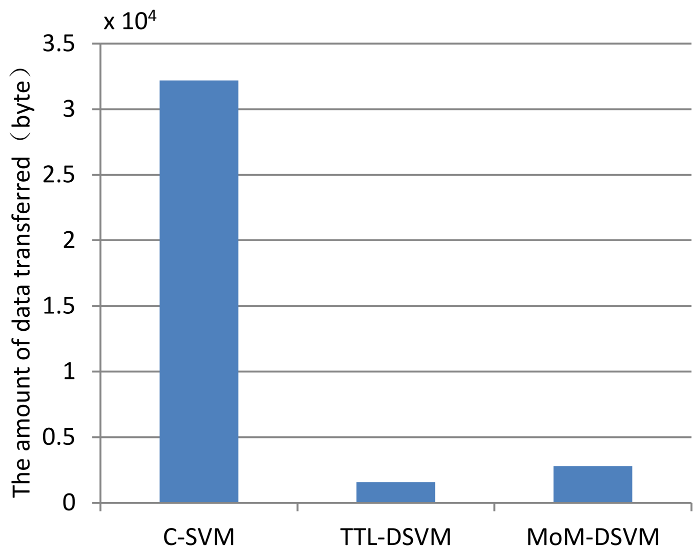

Fig. 2. The amount of data transferred for C-SVM, TTL-DSVM and MoM-DSVM in Synthetic Data

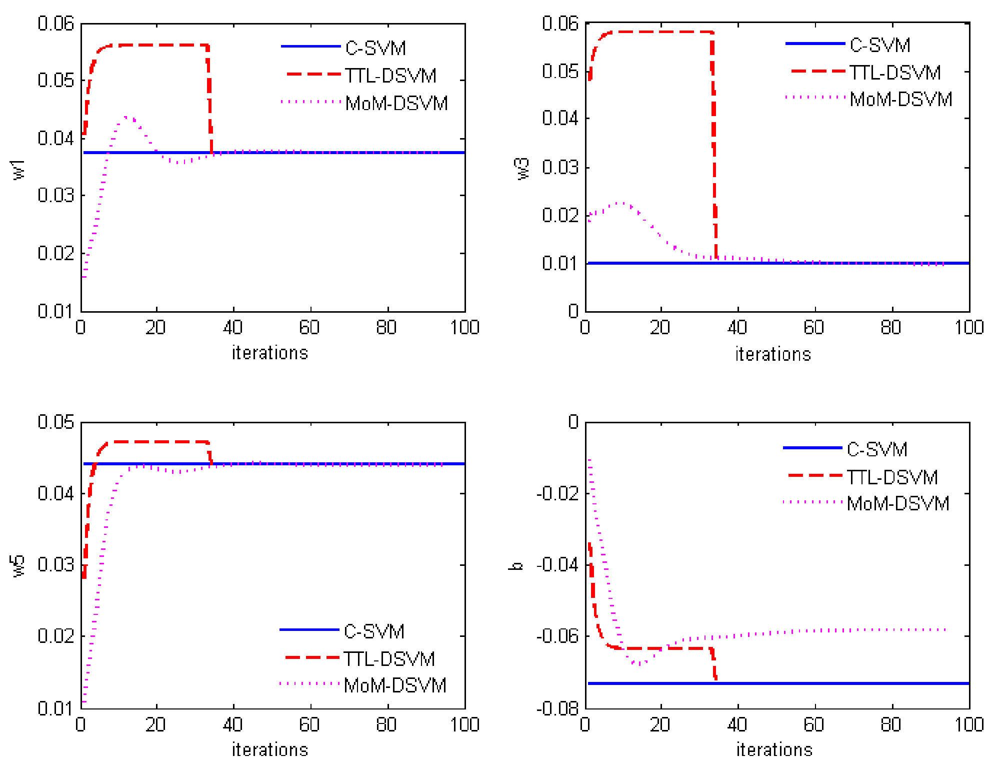

Fig. 3. The iterative processes and convergent results of components of w^* and b^* for C-SVM, TTL-DSVM and MoM-DSVM in Wine Quality dataset

In Figure 4, the amount of data transferred for TTL-DSVM is 82.39% less than that for C-SVM, and 33.68% less than that for MoM-DSVM.

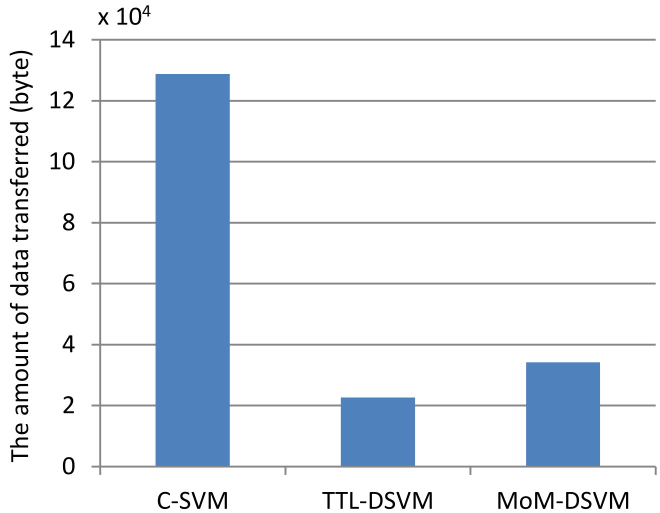

Fig. 4. The amount of data transferred for C-SVM, TTL-DSVM and MoM-DSVM in Wine Quality dataset

4. Conclusion

This work developed a novel distributed training algorithm for linear SVM in WSN, which is inspired by the idea of the local computation and global communication in distributed algorithms and uses the parallel optimization techniques. The proposed algorithm TTL-DSVM divides the training process of SVM into two phases. In each phase, each node only exchanges the parameters of its local classifier with its all direct neighbour nodes. Simulation results show that under the same experimental conditions, the convergent results of TTL-DSVM is almost the same as those of C-SVM, especially, TTL-DSVM has an obvious advantage in the amount of data transferred compared with C-SVM and MoM-DSVM. TTL-DSVM algorithm provides an effective distributed training approach for linear SVM in WSN. In the future work, TTL-DSVM will be extended to solve the nonlinear SVM problems in WSN.

Acknowledgements

This research was financially supported by the National Natural Science Foundation of China under Grant No. 61203377.

References

Aswathy et al. (2012). Security Enhancement in Wireless Sensor Networks using Machine Learning. *IEEE 14th Int. Conf. on High Performance Computing and Communications*, 1264-1269.

Bertsekas, Tsitsiklis. (1997). Parallel and Distributed Computation: Numerical Methods. *Athena Scientific*.

Flouriet al. (2006). Training a Support-Vector Machine-Based Classifier in Distributed Sensor Networks. *14th European Signal Processing Conference*, 1-5.

Flouriet al. (2009). Optimal Gossip Algorithm for Distributed Consensus SVM Training in Wireless Sensor Networks. *16th International Conference on Digital Signal Processing*, 886-891.

Foreroet al. (2010a). Consensus-Based Distributed Linear Support Vector Machines. *9thACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 35-46.

Foreroet al. (2010b). Consensus-Based Distributed Support Vector Machines. *Journal of Machine Learning Research*, *11*, 1663-1707.

Kim et al. (2012). Multi-Target tracking using Distributed SVM Training over Wireless Sensor Networks. *IEEE International Conference on Robotics and Automation*, 2439-2444.

Lu et al. (2008). Distributed Parallel Support Vector Machines In Strongly Connected Networks. *IEEE Transactions on Neural Networks*, *19*(7), 1167－1178.

Liu et al. (2013). Application of a Modified SVM Multi-Class Classification Algorithm in Wireless Sensor Networks. *Journal of China University of Metrology*, *24*(3), 298-303.
Sardellittiet al. (2010). Fast Distributed Average Consensus Algorithms Based On Advection-Diffusion Processes. *IEEE Transactions on Signal Process*, *58*(2), 826-842.
Scholkopf, Smola. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.
Shahidet al. (2012). Quarter-Sphere SVM: Attribute and Spatio-Temporal Correlations based Outlier & Event Detection in Wireless Sensor Networks. *IEEE Wireless Communication and Networking Conference*, 2048-2053.
Wang et al. (2009). Support Vector Machine for Distributed Classification: A Dynamic Consensus Approach. *IEEE/SP 15th Workshop on Statistical Signal Processing*, 753 - 756.
Cortez et al. (2009). Modeling Wine Preferences by Data Mining From Physicochemical Properties. *Decision Support Systems*, *47*(4), 547-553.