Neural Network and SVM classification via Decision Trees, Vector Quantization and Simulated Annealing

JOHN TSILIGARIDIS
Mathematics and Computer Science Department
Heritage University
Toppenish, WA
USA
tsiligaridis_j@heritage.edu

Abstract: - This work provides a method for classification using a Support Vector Machine (SVM) via a Decision Tree algorithm and with Vector Quantization. A probabilistic Decision Tree algorithm focusing on large frequency classes (DTPL) is developed. A method for SVM classification (DT_SVM) using Tabu Search (TS) via DTs is developed. In order to reduce the training complexity of the Support Vector Machine (SVM), the DTPL performs partitions that can be treated as clusters. The TS algorithm can provide the ability to approximate the decision boundary of an SVM. Based on DTs, a SVM algorithm is developed to improve the training time of the SVM considering a subset of the cluster's instances. To reduce the SVM training set size a vector quantization algorithm (the LBG) is used. The LBG classifier is based on Euclidean Distance. Finally, an optimization method, the Simulated Annealing (SA), is applied over the quantization level for discovering of a minimization criterion based on error and low complexity to support the SVM operation. The V_S_SVM can provide lower error at a reasonable computational complexity.
A Neural Network (NN) is composed of many neurons that are linked together according to a specific network topology. Main characteristics of SVM and NN are presented. Comparison between NN and SVM with two types of kernels show the superiority of the SVM. The V_S_SVM with RBF kernel can be compared with DT_SVM and provide useful results. Simulation results for all the algorithms with different complexity data sets are provided

Key-Words: - SVM, Neural Networks, LBG , Decision Trees , Simulated Annealing, Data Mining

1 Introduction

Data Mining includes many significant methods that can be used cooperatively and supportively to improve prediction accuracy over any classification problem. DTs are one of the most popular techniques of Data Mining [1],[2],[3],[4],[5]. They use a tree structure to represent a partition of the space. A DT algorithm, based on classes with large frequency, is created from data using probabilities (DTPL). A theorem is developed for discovering a complete DT and the identification of don't care attributes. To avoid repetition or replication the criterion of elimination of a branch is also applied.
For classification purpose to reduce the training time of SVM using the entire data, the DT_SVM algorithm based on DTPL and TS is developed.
TS is a heuristic procedure for solving optimization problems [6]. TS works with a set of restrictions. The tabu list contains the forbidden moves. A certain set of moves and aspiration criteria are used [7]. An iterative learning algorithm based on sample selection called "SVC" is developed in [8]. In [9] a Clustering-Based (CB_SVM) is developed for hierarchical clustering of large data set.
In many of the real world classification applications the interest is concentrated in the use of NNs [10],[11]. The clusters that are created from the DTs can be linearly separable. In this case, majority voting is employed. This involves converting a node N into a leaf and labeling it with the most common class in D.
The training process is time consuming and the Vector Quantization (VQ) is used in order to decrease this time. The VQ is a lossy data compression method which is based on the principle of a block encoding. Given a vector source, a distortion measure and the number of codevectors, a codebook and a partition of the space are discovered providing smaller average distortion. Thre are two criteria for VQ; the nearest neighbor and the centroid. The LBG algorithm [12] can follow the two optimal criteria and working iteratively can provide a codebook S' from training set S with much smaller size.

After the LBG quantization, and in order to achieve low error with a number of support vectors (in %) the SA algorithm is applied. SA is capable of escaping from local optima [16]. The key algorithmic feature of SA is that it provides a means to escape of local optima by allowing hill-climbing moves[16] (i.e. moves which worsen the objective function value). The SA can also be used also for the Traveling salesman problem (TSP) [17].

The V_S_SVM represents the second layer after the construction of the example prototypes with the use of the LBG classifier and can provide considerable solution following the predefined criterion of the low error and number of support vectors.

The paper is organized as follows. Section 2 contains the DTPL description, Section 3 and 4 have the SVM and TS description respectively. Section 5 and 6 includes the DT_SVM description and NN information. Section 7 and 8 include the SA and the LBG algorithms respectively. Section 9 is referred to the V_S_SVM.

Finally, simulation results are provided in Section 10.

2 DTPL

DTs split the input space into hyper-rectangles according to the target. The DT classifier is a method that can be used as a preprocessing step for SVM. The DTPL can be created in the following phases:

Phase 1: Discover the root (from all the attributes)

$$P(EA) = \sum_{C} \sum_{A} p(A) * p\left(\frac{C}{A}\right)$$

where A : the attributes of the tuples and C the classes (attribute test). MP = max (P(EA)) //max attribute test criterion

Phase 2: Split the data into smaller subsets, so that the partition to be as pure as possible using the same formula. The measure of nodes impurity is the MP. Continue until the end of the attributes.

The large frequency classes (DTPL) are also extracted. The CEB criterion eliminate redundant branches. Most of the decision trees inducers require rebuilding the tree from scratch for reflecting new data that has become available.

For an attribute (attr1) with value v1 , if there are tuples from attr2 that have all the values in relation with v1 (of attr1) then the attr2 is named as: *do n't care* attribute. The criterion of elimination of Branch (CEB) is used to avoid the repetition and the replication and it is given by:

$$P_{CEB} = p(A_1 = a_1, \dots, A_{|A|} = a_{|A|} | C = c_i) = \prod_{i=1}^{|A|} p(A_i = a_i | C = c_i)$$

If the $P_{CEB}= 0$, between two attributes (A1, A2) then A2 is don't care attribute. The CEB criterion is valid when $P_{CEB} \neq 0$. CEB examines the cases of extensions for all probable partitions of an attribute, to avoid repetitions or replications of attributes.

Theorem: the CEB criterion can determine the existence of a small DT with the best accuracy (100%, or complete) avoiding repetitions and replications. *Proof:* Because, when the CEB criterion is valid, discourages the repetition.

The DTPL is used for the generation of the clusters from the leaves.

3 SVM

The SVM is a classification method that can provide better accuracy for some other methods and it has been applied in many areas [8],[9] support vector machines (SVMs), a method for the classification of both linear and nonlinear data. In a nutshell, an SVM is an algorithm that works as follows. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a "decision boundary" separating the tuples of one class from another). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors ("essential" training tuples) and *margins* (defined by the support vectors) [1].

SVM creates a line or a hyper-plane between two sets of data for classification. SVM performs best among current classification techniques, due to its ability to capture non-linearities.

The learning process involves optimization of a cost function which is provably convex. This contrast with neural network approaches where the exist of false local minima in the error function can complicate the learning process. Unlike neural network approaches the architecture is determined by the algorithm and not found by experimentation.

With different kernel functions may result in different performances. Two types of kernel functions for SVM are presented; the linear and the

polynomial. Both of them are used in our experiments.

4 TS

The TS [6],[7] works using neighbourhood structures, and it utilizes a short term memory structure called a tabu which is essentially a list of forbidden moves or solutions. For the TS problem, we consider the well-known one-flip neighbourhood. Similarity measure is now based on the Euclidean norm so that points closest to each other in Euclidean space are grouped under one and only one cluster. The TS can find the boundary instances that belong to adjacent classes and have the minimum distance. The closest pair can be defined as the min (x_i -x_j), where x_i Є C_i and x_j Є C_j. These pairs (p_{ij}) will be separated from all the other instances ($rest_i$ and $rest_j$) of the two adjacent clusters. Randomly selected instances (ran_i) from $rest_i$ (of Ci) are also included in the SVM training.

In a tabu list for C_i the forbidden moves (instances of C_i – p_{ij} – ran_i) are included. The final list of Ci will contain the closest pairs (p_{ij}) of two adjacent clusters and the random selected instances (ran_i). For example, if the total instances of C_i are 20, the closed pairs are 5 (p_{ij}) and the random selected instances from C_i are 3 (ran_i) then the forbidden moves are: 12 (20-5-3). The total instances for the training SVM will be 8 instead of 20.

The TS is used to support the search for finding boundaries for SVM. Instances closer to decision boundaries are the most important for SVM.

5 DT_SVM

For each cluster C_i, a subset of instances, S_i, are selected: the boundary points between two adjacent clusters (C_i and C_j) and other randomly selected instances from the C_i. Instead of the entire set of instances of a cluster C_i , this subset will be used for SVM training.

The proposed method (DT_SVM) has the following steps:

1.Use DTPL for the training set and receive the leaves which contain instances in clusters.

2. Gather all the leaves of DTPL and use them as clusters.

3. From a cluster prepare a subset of instances containing the closest pairs (CP) of instances that belong to adjacent clusters -using TS- and a randomly selected instance from the rest of instances of the cluster.

4. SVM training.

The minority class instances (examples) from each leaf can be removed in order to have pure clusters.

6 NN

A NN is a collection of units that are connected in some pattern to allow communication between the units. The test data set and the training data set should be disjoint (so that test data are not used during training). For the NN in the input layer the neurons correspond to prediction attribute values of the data set, and the output layer represents the predicted classes [10],[11].

The specification of a typical neural network model requires the choice of the type of inputs, the number of hidden units, the number of hidden layers and the connection structure between the inputs and the output layers.

The main characteristic of neural networks (NN) is their ability to generalize information, as well as their tolerance to noise. NNs are often regarded as black boxes since their predictions cannot be explained clearly. The performance of NNs is sensitive to the specific architecture used to arrange the computational units.

NN that are non-linear statistical data modelling tools can be used to model complex relationships between inputs and outputs or to find patterns in data. Therefore, for a number of items in different classes, the NN can learn to classify items. It takes a while to learn, but then it can instantly classify new inputs.

7 Simulated Annealing

Simulated Annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. At each step the SA considers some neighboring state s' of the current state s, and probabilistically decides between moving the system to the state s' or staying in state s. These probabilities ultimately lead the system to move to states of lower energy. This step is repeated until the system reaches a state that is good enough for the application [14],[16]. SA belongs to metaheuristics and uses neighbors of a solution as a way to explore the solution space and also accepts worse neighbors in order to avoid getting stack in local optima. In this way it can find the global optimum if run for a long enough amount of time. The acceptance probability is important in avoiding entrapment in a local optimum depending on the case of the new

solution (better or worse) [14]. If the new solution is better, then the current solution is updated with the new one. On the other hand, if the new solution is worse, then replacement by a generated neighbor is accepted by a certain probability.

8 Vector Quantization

Vector quantization is a problem that given a vector source with its statistical properties known, given a distortion, and given the number of codevectors , finds the distortion and a partition of the space which result in the smallest average distortion [13].
A vector quantizer maps k dimensional vectors in the vector space R^k into a finite set of vectors Y= $\{y_i : i=1,2,..N\}$. Each vector y_i, is called a code vector or a codeword, and the set of all the codewords is called a codebook. Associated with each codeword, y_i, is a nearest neighbor region called Voronoi region, defined by: $V_i = \{x \in R^k : \|x-y_i\| \leq \| x-y_j\|$, for all $j \neq i$.

The design problem of the codebook can be formulated as follows. Given the training set and the number of codevectors , the codevectors and the partition space are discovered so that the squared-error distortion measure (the average distortion) will be minimized. For this purpose two are the criteria: the nearest neighbor and the centroid condition.

The LBG an iterative algorithm [13] can solve the two above optimality criteria. It takes a set of input vectors S= $\{x_i \in R^k , i=1,..,n\}$ and create s an output a set of representatives of vectors C= $\{c_j \in R^k ,j=1,..,M\}$ and M < n.
An LBG pseudocode:
1. input training set (vectors) S.
2. initiate a codebook C
3. initialize $D_0=0$, k=0
4. classify the n training set into K clusters

5. update cluster centers cj , j=1…,K by

$$\boldsymbol{c}_j = \frac{1}{|S_j|} \sum_{\boldsymbol{x}_i \in S_j} \boldsymbol{x}_i$$

6. k<- k+1 , compute the distortion :

$$D_k = \sum_{j=1}^{K} \sum_{\boldsymbol{x}_i \in S_j} \left\| \boldsymbol{x}_i - \boldsymbol{c}_j \right\|_p$$

7. If $\frac{D_{k-1}-D_k}{D_k} > \epsilon$

the steps 4-6 are repeated
else

get the codebook :

$$C = \{\boldsymbol{c}_j \in R^d | j = 1,2,\dots,K\}$$

9 V_S_SVM

Let $\{\mathbf{x}_I, y_i\}$ for $1 \leq i \leq m$ is a set of trining examples , $\mathbf{x}_i \in R^n$ which belong to a class labeled by $y_i \in \{-1,+1\}$.

From an SVM the decision function [15] comes from:

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i^* g_i k(x_i, x) + b) \quad (1)$$

And the coefficient α_i^* come form the maximization of the function:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j g_i g_j k(x_i, x_j)$$

With the constraints:

$\sum_{i=1}^{m} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$ (regularization parameter)

The coefficient α_i^* defines a maximal margin hyperplane in a high-dimensional feature space where the data are mapped through a non linear function φ ($\varphi: R^d \rightarrow F, F$ is a high dimensional feature space). The mapping φ is performed by a kernel function $K(\mathbf{x}_i , \mathbf{x}_j)$ that defines an inner product in this feature space such that $\varphi(\mathbf{x}_i).\varphi(\mathbf{x}_j) = K(\mathbf{x}_i , \mathbf{x}_j)$. This formulation of the SVM optimization problem is the hard margin formulation. Every training point satisfies the inequality $y_i f(\mathbf{x}_i) \geq 1$ and for points $\mathbf{x}_i$ with corresponding $\alpha_i > 0$ an equality is satisfied. These points are called support vectors [15].

We propose a new technique for SVM-based classification that is based on optimized vector quantization, with the goal of minimizing training time and complexity. The training data is first quantized using the LBG and a kernel SVM is trained on the quantized data. For SVM the RBF kernel is used. The dataset is separated into the training set (T) and the testing (S). The simulated annealing method is used to optimize over the quantization level in order to achieve low error at a low complexity. The criterion being optimized is a linear combination of the test error and the number of support vectors.

The average error is computed for the decision functions $f(S)$ (1) of the testing set S considering the regularization parameter C and the σ (kernel parameter) constant. The initial value of $\alpha = 2$ $(2^a, codevectors)$. The SVM works iteratively for increasing α values. Also, the minimum number of support vectors and the error (in %) are computed from the decision function, after increasing the α values.

The simulated annealing algorithm starts from a quantization level, picks a random neighbouring quantization level, and evaluates the criterion; if the objective is improved the new level is accepted; otherwise a probabilistic acceptance decision is made and the process repeats.

Simulation results show that the algorithm achieves low error at a reasonable computational complexity.

10 Simulation

Five scenarios have been developed:

Scenario 1: (*ID3 vs DTPL):* From Figure 1, DTPL has lower value than the ID3 due to the algorithm construction.

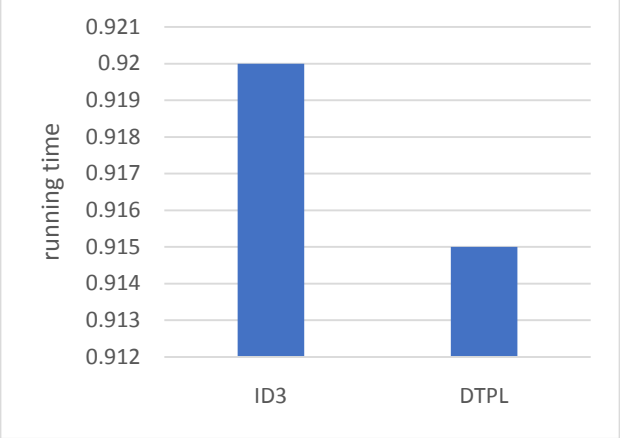

Fig. 1 ID3 vs DTPL

Scenario 2: (DT_SVM vs SVM) From Figure 2 it is shown that DT_SVM has better training time and accuracy than SVM. SVM needs much more training time for all the data.

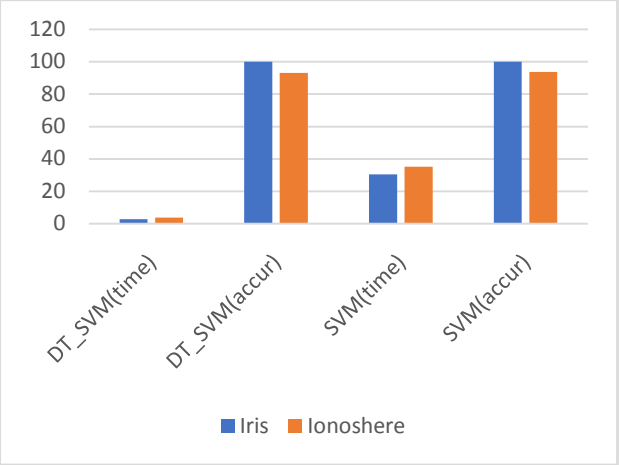

Fig. 2 DT_SVM vs SVM

Scenario 3: (SVM vs NN) NN is competitive to SVM. Depending on the kernel type different results are produced. From Figure 3, using the MNIST dataset, SVM with polynomial; kernel has better classification rate than the NN. NN outperforms the SVM linear kernel

.

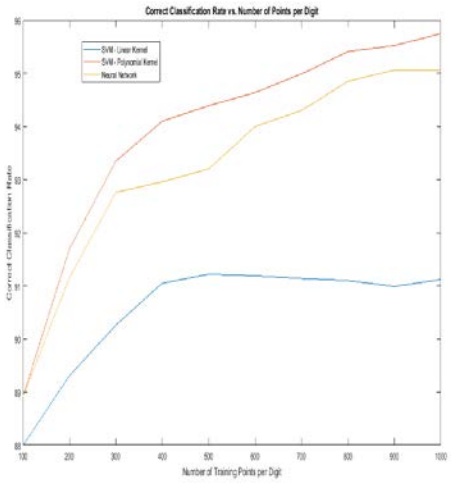

Fig. 3 SVM vs NN

Scenario 4: NN training. From Figure 4 the NN, according to MSE values, has the best training performance at 171 Epochs

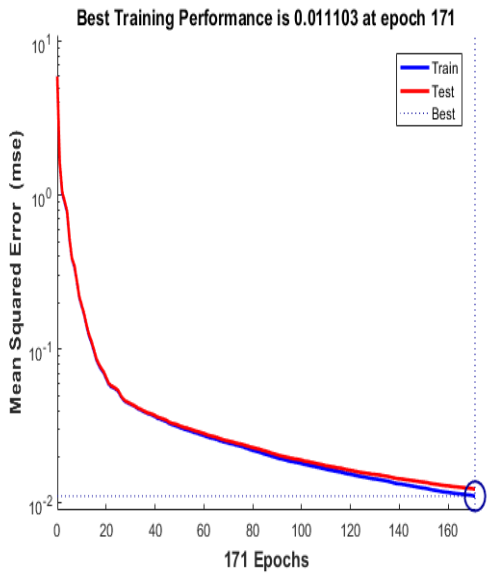

Fig. 4 NN best performance

Scenario 5: (DT_SVM vs CV_SVM) In Fig. 5 it is shown that CV_SVM has better training time than DT_SVM. This is because the LBG diminish the training time. In addition, DT_SVM has better accuracy than CV_SVM because the last use the codevectors.

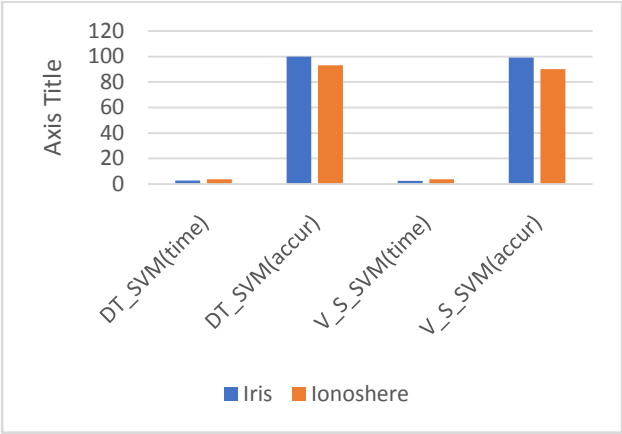

Fig. 5 DT_SVM vs CV_SVM

11 Conclusion

The SVM is computational infeasible on large data sets due to time and space complexities ($O(n^2)$). Instead of applying SVM to the whole data set SVM can be applied to each disjoint region discovered by DTPL. The DT_SVM shoes better performance than SVM. The DTPL provides the partitions (regions) from which the DT_SVM will get a subset with the decision boundaries of adjacent clusters using TS and random selected instances. The V_S_SVM, following the minimization criterion of error and number of support vectors can provide competitive results against the DT_SVM. Finally, SVM with the appropriate kernels outperform the NN. Future work could be based on NN and Genetic Algorithms.

References:

[1] J.Han, M.Kamber,J.Pei, *Data Mining Concepts and Techniques*, Morgan Kaufman,3 ed. 2012.

[2] U.Fayyad, G.Piateski-Shapiro, *From Data Mining to Knowledge Discovery*, MIT Press 1995.

[3] M.Karntardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, IEEE Press,2003

[4] M. Bramer, *Principles of Data Mining*, Springer-Verlag, London Limited, 2007.

[5] L. Rokach, O. Maimon, *Data Mining with Decision Trees: Theories and Applications*, Word Scientific ,2008.

[6] F. Glover,"Tabu Search: A tutorial", https://www.ida.li.se/~zebpe83/heuristic/papers/TStutorial.pdf

[7] J.Gaast, C.Rietveld, A. Gabor, Y. Zhang, A Tabu search Algorithm for application placement in computer clustering, *Computers & Operations Research*, Elsevier, 2014, pp:38-46

[8] Z.Chen, B.Liu, X.He, " A SVC iterative learning Algorithm based on sample selection for large samples", *6th Intern. Conference on Machine Learning and Cybernetics*, Hong Kong, 2007

[9] H.Yu, J. Yang, J.Han, " Classifying Large data Sets using SVMs with Hierarchical Clusters", *SIGKDD 2003*, Washington DC, USA

[10] S, Haykin, *Neural Networks: A comprehensive Foundation*, Pearson, 2ed, 2005.

[11] Fausett L. ,*Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice Hall, NJ, 1994

[12] A.Gersho, R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic,1991.

[13] W. Steeb, *Mathematical Tools in Signal Processing with C++ & Java Simulations*, International School for Scientific Computing

[14] https://en.wikipedia.org/wiki/simulated_annealing

[15] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing Multiple Parameters for

Support Vector Machines, *Machine Learning*, 46, 131-159,2002

[16] D. Henderson, S. Jacobson, A. Johnson, The Theory and practice of Simulated Annealing, *Handbook of Metaheuristics*, Springer, pp.287-319, 2003

[17] A. Anagnostopoulos, L. Michel, P. Henternryck, Y. Vergados, A simulated annealing approach to the traveling tournament problem, *Journal of Scheduling*, Springer, Vol. 9, Issue 2, April 2006, pp 177-193