International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.2, No.10, October 2013 DOI:10.15693/ijaist/2013.v2i10.18-22

Hybrid Classification Model for Multi Class Gene Classification

Henry Alexander. I
Research Scholar
Karpagam University, Coimbatore, India

Dr. Mallika. R
Assistant Professor/Department of Computer Applications
CBM College, Coimbatore

***Abstract*— The main objective of this study is to compare SVM (Structured Vector Machine) classification algorithm and Neural Networks classification algorithms identify the pitfalls and propose a new classification algorithm which is reliable, fast, efficient and robust handling large sample data. The study uses this hybrid algorithm for multi class gene classification**
Generally most of the classification algorithms are working good for small and moderate data, while going for large datasets, the efficiency drops, this study analyses all these factors and proposing a new hybrid model, which solves all theses pitfalls.

Index Terms: SVM, Neural Networks.

I. INTRODUCTION

A .What is Datamining?
Datamining sometimes called data or knowledge discovery) is a process of analyzing data from different perspective and summarizing the result into useful information.

B. What is classification?

- maps data into predefined groups or classes
    - Supervised learning
    - Pattern recognition
    - Prediction

What is Clustering?
Group's similar data together into clusters.

- Unsupervised learning
- Segmentation
- Partitioning

What is Regression?
Is used to map a data item to a real valued prediction variable.

Frequently Used Classification Algorithms.

- Distance Vector Algorithm
- Rot Boost Ensemble Technique
- Simple Bayesian Classifier
- Support Vector Machine For classification
- Decision Tree Based Algorithm
- Back Propagation
-

Frequently Used Clustering Algorithms

- K-means
- Fuzzy C-means
- Hierarchical clustering
- Mixture of Gaussians

II. RELATED WORK

Support Vector Machines (SVM) and kernel related methods have shown to build accurate models but the learning task usually needs a quadratic programming, so that the learning task for large datasets requires big memory capacity and a long time.

In recent years, real-world databases increase rapidly (double every 9 months . So the need to extract knowledge from very large databases is increasing. Knowledge Discovery in Databases (KDD can be defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Data mining is the particular pattern recognition task in the KDD process. It uses different algorithms for classification, regression, clustering and association. We are interested in SVM learning algorithms proposed by Vapnik because they have shown practical relevance for classification, regression and novelty detection. Successful applications of SVMs have been reported for various fields, for example in face identification, text categorization and bioinformatics. The approach is systematic and properly motivated by statistical learning theory. SVMs are the most well known algorithms of a class using the idea of kernel substitution. SVM and kernel-based methods have become increasingly popular data mining tools. In spite of the prominent properties of SVM, they are not favourable to deal with the challenge of large datasets. SVM solutions are obtained from quadratic programs (QP), so that the computational cost of an SVM approach is at least square of

the number of training data points and the memory requirement making SVM impractical. There is a need to scale up learning algorithms to handle massive datasets on personal computers (PCs). The effective heuristics to improve SVM learning task are to divide the original QP into series of small problems incremental learning updating solutions in growing training set, parallel and distributed learning on PC network or choosing interested data points subset (active set) for learning boosting of SVM based on sampling techniques for scaling up learning.

III. Objectives & Overview Of The Proposed Mechanism

A. Objectives

We have created a new algorithm that is very fast for building incremental, parallel and distributed SVM classifiers. It is derived from the finite Newton method for classification proposed by Mangasarian . The new SVM algorithm can linearly classify two million datapoints in 20-dimensional input space into two classes in some seconds on ten PCs (3 GHz Pentium IV, 512 MB RAM, Linux). We briefly summarize the content of the paper now. In section 2, we introduce the finite Newton method for classification problems. In section 3, we describe how to build the incremental learning algorithm with the finite Newton method. In section 4, we describe our parallel and distributed versions of the incremental algorithm. We present numerical test results in section 5 before the conclusion in section 6. Some notations are used in this paper. All vectors will be column vectors unless transposed to row vector by a *T* superscript. The inner dot product of two vectors, *x, y* is denoted by s*x.y*. The 2-norm of the vector *x* will be denoted by ||*x*||. The matrix *A[mxn]* will be *m* data points in the *n*-dimensional real space *Rn*. The classes *+1, -1* of *m* data points are denoted by the diagonal matrix *D[mxm]* of *-1, +1*. *e* will be the column vector of *1*. *w, b* will be the coefficients and the scalar of the hyper-plane. *z* will be the slack variable and *C* is a positive constant. *I* denote the identity matrix.

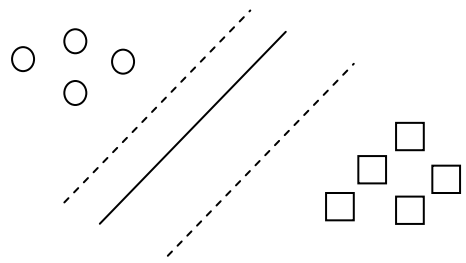
B. Overview of the proposed Mechanism

We propose a new hybrid model which is a combination of Support Vector Machine and Neural Networks. We classify the data using SVM and test the dataset with Neural Networks. We have created a new SVM algorithm which is fast and efficient and helps to classify larger datasets. The remaining information focuses on the new SVM algorithm.

IV. Modified Support Vector System

A. Mobility Oriented Trust System (MOTS)

xT.w – b = -1 xT.w – b = 0



xT.w – b = +1

margin = 2/||w||

Figure 1. Linear separation of the data points into two classes

Let us consider a linear binary classification task, as depicted in figure 1, with *m* datapoints in the *n*-dimensional input space *Rn*, represented by the *mxn* matrix *A*, having corresponding labels *±1*, denoted by the *mxm* diagonal matrix *D* of *±1*.

For this problem, the SVM algorithms try to find the best separating plane, i.e. furthest from both class *+1* and class *-1*. It can simply maximize the distance or margin between the supporting planes for each class (*xT.w – b = +1* for class *+1*, *xT.w – b = -1* for class *-1*). The margin between these supporting planes is *2/||w||* (where ||*w*|| is the 2-norm of the vector *w*). Any point *xi* falling on the wrong side of its supporting plane is considered as an error (having corresponding slack value *zi > 0*). Therefore, a SVM algorithm has to simultaneously maximize the margin and minimize the error. The standard SVM formulation with a linear kernel is given by the following QP (1):

$$\textbf{Min} \quad f(w, b, z) = CeTz + (1/2)\|w\|2$$
$$\textbf{s.t.} \quad D(Aw - eb) + z \geq e \qquad (1)$$

where slack variable $z \geq 0$, constant $C > 0$ is used to tune errors and margin size. The plane (w, b) is obtained by the solution of the QP (1). Then, the classification function of a new datapoint x based on the plane is: $predict(x) = sign(w.x - b)$ SVM can use some other classification functions, for example a polynomial function of degree d, a RBF (Radial Basis Function), or a sigmoid function. To change from a linear to nonlinear classifier, one must only substitute a kernel evaluation in (1) instead of the original dot product. More details about SVM and others kernel-based learning methods can be found in and . Recent developments for massive linear SVM algorithms proposed by Mangasarian reformulate the classification as an unconstrained optimization. By changing the margin maximization to the minimization of $(1/2)\|<w, b>\|2$ and adding with a least squares 2-norm error, the SVM algorithm reformulation with linear kernel is given by the QP (2)

$$\textbf{Min } f(w, b, z) = (C/2)\|z\|2 + (1/2)\|<w, b>\|2$$
$$\textbf{s.t. } D(Aw - eb) + z \geq e \qquad (2)$$

where slack variable $z \geq 0$, constant $C > 0$ is used to tune errors and margin size. The formulation (2) can be rewritten by substituting for $z = (e - D(Aw - eb))+$ (where $(x)+$ replaces negative components of a vector x by zeros) into the objective function *f*. We get an unconstrained problem :

$$\textbf{Min } f(w, b) = (C/2)\|(e - D(Aw - eb))+\|2 + (1/2)\|<w, b>\|2 \qquad (3)$$

By setting *[w1 w2 … wn b]T* to *u* and *[A -e]* to *H*, then the SVM formulation (3) is rewritten by :

$$\textbf{Min } f(u) = (C/2)\|(e - DHu)+\|2 + (1/2)uTu \qquad (4)$$

Mangasarian has shown that the finite stepless Newton method can be used to solve the strongly convex unconstrained minimization problem . The algorithm can be described as the algorithm 1.

Δf(u) Hessian matrix or **Hessian** is a square matrix of second-order partial derivatives of a function. It describes the local curvature of a function of many variables.

δ^2f(u) Gradient of a scalar field is a vector field that points in the direction of the greatest rate of increase of the scalar field.

Flow of the work

- **Selection of Databases (Diabetes, Cancer.**
- **Use 60% of data for training, 40% for testing**
- **Train the data using modified Newton stepless SVM algorithm**
- **Test for accuracy.**
- **Input the trained data to Neural Networks.**

Gradient and Hessian are calculated for all vector elements for minimize the error until the gradient is zero

- Input: training dataset represented by *A* and *D* matrices
- Starting with $u0 \in Rn+1$ and $i = 0$
- Repeat

1) $ui+1 = ui - \delta^2 f(ui) - \Delta f(ui)$

2) $i = i + 1$

Until $\Delta f(ui) = 0$

- Return *ui*

Calculation of gradient *f* at *ui*,

$\Delta f(ui) = C(-DH)T(e - DHui)+ + ui$ **(5)**

Calculation of generalized Hessian of *f* at *ui*,

$\delta^2 f(ui) = C(-DH)Tdiag([e - DHui]*)(-DH) + I$ **(6)**

with *diag([e – DHui]*)* denotes the *(n+1)x(n+1)* diagonal matrix whose *jth* diagonal entr[illegible]nction $(e - DHui)_+$

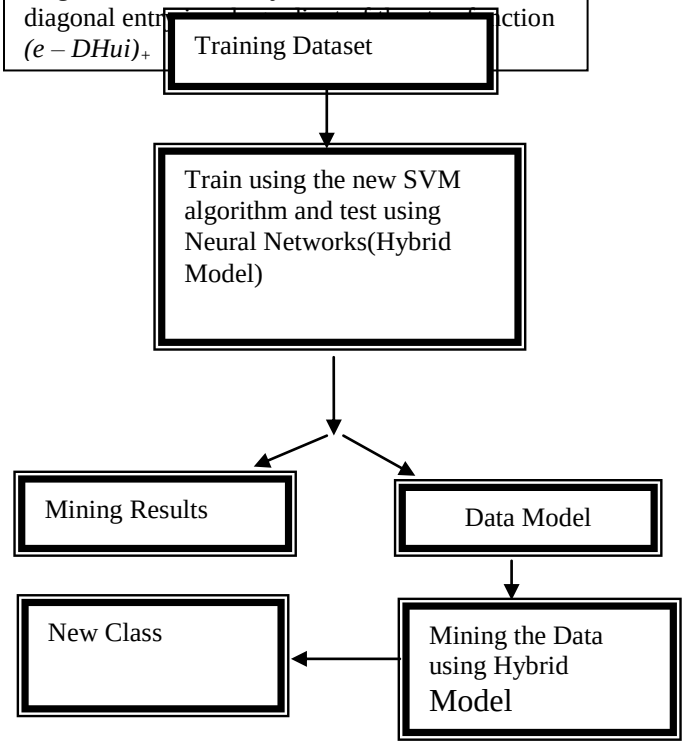


Figure 2. Proposed Model

As mentioned, the input data (training data) is acquired preprocessed and sent as input to the new Hybrid Model, a combination of SVM (Structured Vector Machine) and Neural Networks models. Data is appropriately mined and produced as mining results i.e forms a new class or used to produce summary of results. If not classified the data gain mined until it is classified.

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a *scaling factor*, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error..

A. New Hybrid Model Algorithm

- Input: training dataset represented by *A* and *D* matrices
- Starting with $u0 \in Rn+1$

Use SVM algorithm and get the required training dataset ui.

Let ui be the input to neural networks

1. Initialize the weights in the network (often randomly)
2. repeat

* for each example ui in the training set do

1. O = neural-net-output(network, ui) ;

Forward pass

3. T = teacher output for ui
4. Calculate error (T - O) at the output

Units (Normally this difference will be negligible when comparing with ordinary Neural Networks Algorithm).

If needed follow the following steps for back propagation

5. Compute delta_wi for all weights from hidden layer to output layer ;

backward pass

6. Compute delta_wi for all weights from input layer to hidden layer ;

backward pass continued

7. Update the weights in the network

* end

8. until all examples classified correctly or Stopping criterion satisfied
9. return (network

Actual Class

B. ***Advantages of this algorithm.***

- Data is well trained since we are using combination of two algorithms.
- Classification is fast and accurate.

	Data1	Data2
Training Set	104	106
Testing Set	103	105
Training time (s) Accuracy (%)%	0.025 97.25	2.585 93.36%

- Able to handle large datasets.
- Weight matrix is minimized.
- No need for much weight adjustment during back propagation.
- We end in a accurate classification
- Error rate is negligible**.**

VI Performance Evaluation

Predicted Class

	C1	**C2**
C1	**True Positive**	**True Negative**
C2	**False Positive**	**True Negative**

Table1. Confusion Matrix

A. Performance Metrics

Sensitivity = t_pos / pos
Specificity = t_n eg / neg
Precision = t_pos / (t_pos + t_neg)
Accuracy = sensitivity * pos / (pos + neg) + specificity * neg / (pos + neg)

Sensitivity (also called the *true positive rate*, or the recall rate in some fields) measures the proportion of actual positives which are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition).

Specifity measures the proportion of negatives which are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition, sometimes called the *true negative rate*).

Accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's actual (true) value.

Precision of a measurement system, also called reproducibility or repeatability, is the degree to which repeated measurements under unchanged conditions show the same results

C. Results

Table2. Result

VII CONCLUSION

Thus proposed model is a combination of SVM model and Neural networks, it has dual advantages of both these models. It removes the drawbacks of SVM model and makes classification accurate, reliable and efficient.

REFERENCES

1. U. Fayyad, D. Haussler, and P. Stolorz, "Mining scientific data," *Communications of the ACM*, Vol. 39, 1996, pp. 51-57.
2. J. Zhang, W. Hsu, and M. L. Lee, "Image mining: Issues, frameworks and techniques," in *Proceedings of the 2nd International Workshop Multimedia Data Mining*, 2001, pp. 13-20.
3 *B*. Nagarajan and P. Balasubramanie,
"Cluttered Background Removal in Static Images with Mild Occlusions" *International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009*
4. C. Ordonez and E. Omiecinski, "Image mining: A new approach for data mining,"
Technical Report GIT-CC-98-12, College of Computing, Georgia Institute of Technology, 1998.
5. B. Nagarajan and P. Balasubramanie, "Neural Classifier for Object Classification with Cluttered Background Using Spectral Texture Based Features"
Journal of Artificial Intelligence, 2008, ISSN 1994-5450
6. A. Vailaya, A. T. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for
Content-based indexing," *IEEE Transactions on Image Processing*, Vol. 10, 2001, pp.117-130.
7. R. F. Cromp and W. J. Cambell, "Data mining of multidimensional remotely sensed images," in *Proceedings of the 2nd International Conference on Information and Knowledge Management*
8. A Simple, Fast Support Vector Machine Algorithm For Data Mining
Hiep-Thuan Do, Nguyen-Khang Pham, Thanh-Nghi Do
College of Information Technology, Cantho University
1 Ly Tu Trong Street, Ninh Kieu District
Cantho City, Vietnam , Fundamental & Applied IT Research Symposium 2005

Authors Profile

I.Henry Alexander received MCA from Madura College affiliated to Madurai Kamarajar University, M,Phil from Manonmaniam Sundaranar University, Tirunelveli, currently pursuing Ph.D from Karpagam University, Coimbatore. University. Topper in both Masters Degree. Having more than eleven years of experience in academic and industry, having worked in Singapore, Dubai and Germany(under guidance of Prof. Reinard Selten Nobel Prize Winner for Game Theory).He has published 3 National and two International Journals

Dr.R.Mallika,India obtained her PhD in Computer Science in 2010 from Mother Theresa University, Kodaikannal. With teaching experience of 12 Yrs and currently associated with CBM College of Arts and Science, Sakethapuri, Kovaipudur, Coimbatore-4 as Assistant Professor in Department of Computer applications.With research experience of 5 years published more than 15 International publications in Conferences and Journals.