ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of DEVELOPMENT RESEARCH

International Journal of Development Research
Vol. 4, Issue, 5, pp. 1092-1096, May, 2014

Full Length Research Article

IMPLEMENTATION OF HYBRID CLASSIFICATION MODEL IN DISTRIBUTED SYSTEMS FOR NETWORK MONITORING

[1*]Henry Alexander, I. and [2]Dr. Mallika, R.

[1]Research Scholar Karpagam University, Coimbatore, India
[2]Assistant Professor, Department of Computer Science, CBM College, Coimbatore

ARTICLE INFO

Article History:
Received 06th February, 2014
Received in revised form
10th March, 2014
Accepted 18th April, 2014
Published online 31st May, 2014

Key words:
SVM, Neural Networks. Kernel,
Gradient, Hessian. Hybrid,
Distributed Systems,
Smart Agents.

ABSTRACT

The main objective of this study is to compare SVM (Support Vector Machine) classification algorithm and Neural Networks classification algorithms identify the pitfalls and propose a new hybrid classification algorithm which is reliable, fast, efficient and robust handling large data sets. A new version of Support Vector Machine algorithm is designed and developed in the study which is used for training the dataset followed by testing using Neural Networks classification algorithm. Generally most of the classification algorithms are working well 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 these drawbacks. The second main objective of this study is to analyze whether the number of back propagation steps are minimized in Neural Networks algorithm. This model can be used to monitor and predict the networking issues that occur in distributed systems.

INTRODUCTION

A. 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. Classification

Maps data into predefined groups or classes

- Supervised learning
- Pattern recognition
- Prediction

C. Clustering

Group's similar data together into clusters.

- Unsupervised learning
- Segmentation
- Partitioning

Corresponding author: Henry Alexander, I.
Research Scholar Karpagam University, Coimbatore, India

D. Regression

Is used to map a data item to a real valued prediction variable.

E. 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

F. Frequently Used Clustering Algorithms

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

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.

Experimental Analysis

Objectives

We have created a new algorithm that is very fast for building incremental, parallel and distributed SVM classifiers. A new kernel function is proposed and there by using this kernel function a new algorithm is proposed and designed to classify larger dataset for accuracy and efficiency. 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.

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.

Modified Support Vector System

a) Design of New Kernel Function

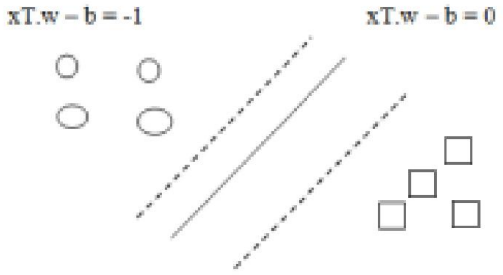

Figure 1. Linear separation of the data points into two classes

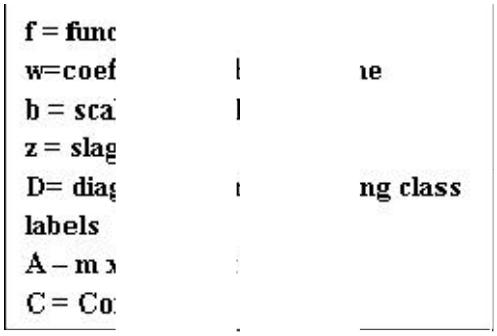

Figure 2. Derivatives used in Kernel Function and Algorithm

For the study consider a binary classification as shown in figure 1, with m data points in the n dimensional input space Rn, with m x n matrix A, having corresponding labels ±*1*, denoted by m x m diagonal matrix D of ±*1*.

The SVM algorithm tries to find the best separating plane, i.e furthest from both class +I and –I.

It cam maximize the distance margin between the supporting planes for each class by the formulas (xT.w-b = +I) for class +I and xT.w-b=-I for class –I. The margin between the separating planes is 2/||w||.(where ||w|| is the 2-norm form of vector w). Ay point falling on wrong side of its support plane is considered to be an error.(having corresponding slag variable zi > 0). There for a SVM has to maximize the margin to minimize the error. The standard SVM formulae for a linear kernel function is given by the following QP(1)

Min $f(w, b, z) = CeTz + (1/2)||w||2$

s.t. $D(Aw - eb) + z \geq e$ (1)

Then, the classification function of a new data point x based on the plane is predict(x) = sign(w.x-b).

If the sign is positive, corresponds to right classification else wrong classification. Frequently used kernel functions and their pitfalls are listed below.

Table 1. Different Kernel Functions and their Drawbacks

S.No	Kernel Function	Drawbacks
1	Gaussian Function	Efficiency drops for larger datasets
2	Polynomial Function	Efficiency drops for larger datasets
3	Radial Basis Function	Efficiency drops for larger datasets
4	Fisher Function	Efficiency drops for larger datasets
5	Graph Function	Efficiency drops for larger datasets
6	String Function	Efficiency drops for larger datasets

where slack variable $z \geq 0$, constant $C > 0$ is used to tune errors and margin size.. The proposed study focuses on a new kernel function which is derived as per the calculations listed below. 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)

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

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

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

$$Min\ f(u) = (C/2)||(e - DHu)+||2 + (1/2)uTu \quad (4)$$

Thus by using the above developed kernel function and correspondingly taking the first order and second order derivatives for the function for each data point, the efficiency and accuracy can be obtained for larger datasets. The algorithm can be described as the algorithm 1.

$\Delta f(u)$ 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.

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

Proposed Model

Preprocessing Steps for the Hybrid Model

- Selection of Databases (Diabetes, Cancer.
- Use 60% of data for training, 40% for testing
- Train the data using 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 entry is sub-gradient of the step function $(e - DHui)+$.

Figure 3. Pseudo Code for Modified Support Vector Machine Algorithm

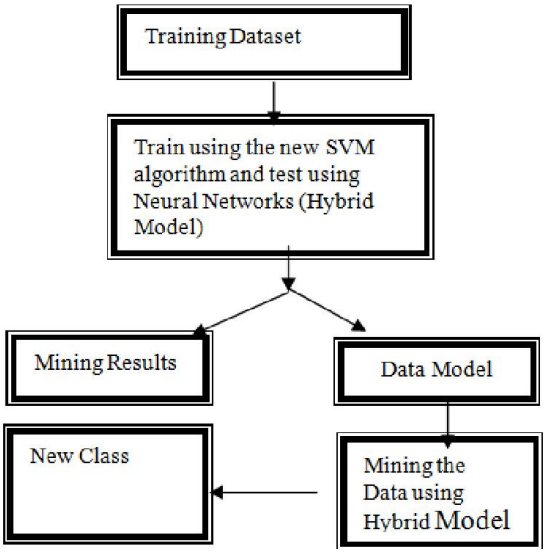

Figure 4. Proposed Model

As mentioned, the input data (training data) is acquired preprocessed and sent as input to the new Hybrid Model, a combination of SVM (Support Vector Machine) and Neural Networks. If the model exists, data is classified using the model else a new hybrid model is designed and accordingly data is trained using SVM and tested using Neural Networks. A new class is generated.

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.

Advantages of this algorithm

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

Implementation of Hybrid Model in Distributed Systems

- Input: training dataset represented by *A* and *D* matrices
- Starting with *u0* € *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

Figure 5. Pseudo Code for Hybrid Algorithm

Distributed System

A distributed system is a system consisting of computers that do not share a common memory or a synchronized clock. The computers in a distributed system are connected via a communications network. In the study classifying various networking issues and labeling the tuples, which helps to analyze and predict the problems which may happen in future. Various predefined networking issues are collected on the following issues

1. Server Crash
2. Network Congestion
3. Cable failure
4. Networking
5. Equipment's fault

Using these predefined data a model is designed. Smart Agents acting on various servers will collect the data

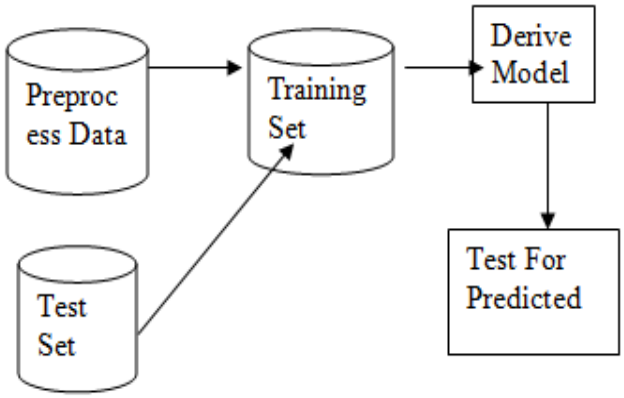

Figure 6. Role of Hybrid Algorithm

Role of this Model

This hybrid model will be installed in every server in distributed environment. Based on the predefined dataset, the model will tests whether any issues in the distributed systems matches the predefined dataset, if it matches the system takes appropriate action to resolve the problems. The model also predicts and forecast whether any networking issues might occur in future and helps to avoid the networking disasters that may occur in future.

Implementation of the Model

- Initialize all the servers and deploy the smart agents
- Develop appropriate routing algorithm to identify the problems in the distributed environment.
- Develop suitable GUI to visualize the problems by the user.
- Use the Hybrid Model to classify the problems**.**
- Based on the problems classified, adopt suitable visualization techniques to display the problems that may occur in future

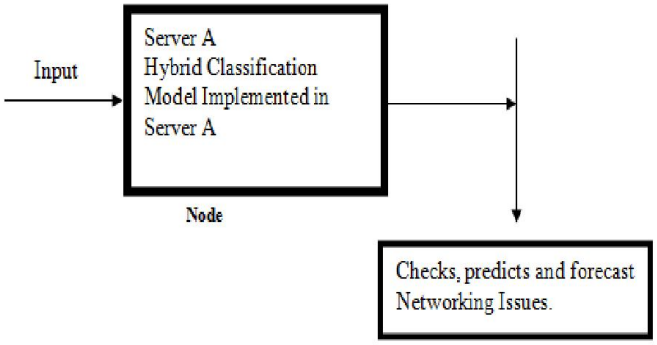

Figure 7. Implementation of Hybrid Algorithm in a single node in Distributed Environment

Performance Evaluation

Table 2. Testing Results for different data size

	Data1	Data2
Training Set	10^4	10^6
Testing Set	10^3	10^5
Training time (s)	0.025	2.585
Accuracy (%)%	97.25	93.36%

Table 3. Confusion Matrix

	Actual Class	Predicted Class
	C1	C2
C1	True Positive	True Negative
C2	False Positive	True Negative

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

RESULTS

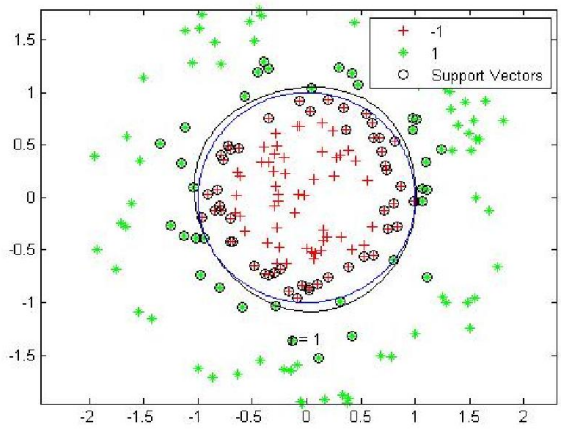

Figure 8. Plotted in Graph Using Hybrid Algorithm

Conclusion

Thus this model helps to solve the problems listed below

- One it removes the drawbacks of conventional Support Vector Machine algorithm using either of the Kernel functions listed in the study.
- The algorithm helps in accuracy and efficiency measures for larger datasets.
- Second advantage of this algorithm is, it minimizes the number of back propagation in Neural Networks algorithm which reduces the time and increases the efficiency in classifying data.
- This algorithm is fast, efficient and accurate for larger datasets.

REFERENCES

1. A Simple, Fast Support Vector Machine Algorithm For Data Mining Hiep-Thuan Do, Nguyen-Khang Pham, Thanh-Nghi Do *College of Information Technology, Cantho University1 Ly Tu Trong Street, Ninh Kieu District Cantho*
2. Improvements to the SMO Algorithm for SVM Regression 3.S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers.
3. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144–152, Pittsburgh, PA, 1992. ACM Press.4.O. Chapelle, *City, Vietnam , Fundamental & Applied IT Research Symposium 2005*
4. U. Fayyad, D. Haussler, and P. Stolorz, "Mining scientific data," *Communications of the ACM*, Vol. 39, 1996, pp. 51-57.
5. 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.
6. *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*
7. 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.
8. 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*
9. 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.
10. 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*
11. 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*

****