SVM: The Qualitative and Quantitative Monolithic Predictor

Abhijit Adhikari [1], **Chetan Nimba Aher** [2]

[1, 2] Pune University, RMD Sinhgad School of Engineering, Pune, India

Abstract: ***Machine Learning is one of the major popular research topics of Artificial Intelligence and its relay with the evolution of techniques and methods which enable the data processor to learn and execute activities. Support Vector Machine (SVM) is a segregated classifier deal with both linear and nonlinear data from hyperplane with the help of Supervised Learning Approach. Whereas Statistical Learning Theory cannot procure location information in a Sentient Computing because of functional dependencies of geographic coordinates from RSSI but SVM can predict the location fingerprint with regression estimation and linear operator inversion and realize the actual risk minimization by structural risk minimization, so that SVM can also obtain a good learning outcome in the face of less sample volume. The basic idea of SVM is for the linearly separable samples, to find the optimal classification hyperplane which can describe accurately the samples into two categories for the linearly nonseparable problems; to transform the linear non-separable problems in the original space into the linearly separable problems in high-dimensional feature space by a nonlinearly transformation for the given samples. SVM gives a very low error rate when used as a classifier.***

Keywords: Support Vector Machine (SVM), Received Signal Strength Identity (RSSI), Hyperplane, Location Fingerprint, Statistical Learning.

1. Introduction

In 1992 Vapnik introduced Support Vector Machine. This uses machine learning theory to maximize predictive accuracy and automatically avoiding over-fit to the data. A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane a labelled training data. The algorithm outputs an optimal hyperplane. Let's consider the following simple problem as shown in figure 1 [2]. For a linearly separable set of 2D-points from one of two classes.

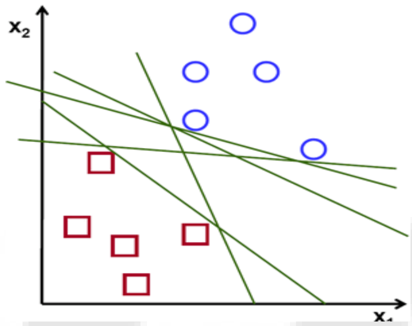

Figure 1: A linearly separable set of 2D-points which belong to one of two classes.

In this case we deal with lines and points in the Cartesian plane instead of hyperplanes and vectors in a high dimensional space. However the same concepts apply to jobs where the examples to classify lie in a space whose dimension is higher than two.

In the above picture we can see that there exist multiple lines that offer a solution to the problem. But we cannot say that any of them better than the others? So we have to define a criterion to estimate the worth of the stocks: A telephone circuit is defective if it gets too close to the points because it will be noise sensitive and it will not generalize correctly. It should be better to set the line passing as far as possible from all points [2].

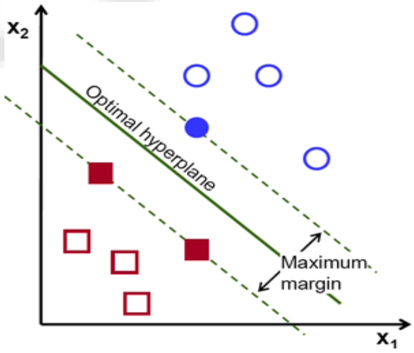

Figure 2: The optimal separating hyperplane *maximizes* the margin of the training data

Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples as shown in figure 2 [2]. This distance receives the important name of margin within SVM's theory. Hence the optimal separating hyperplane maximizes the margin of the training data.

1.1 Statistical Learning Theory

The statistical learning theory provides a framework for studying the problem of gaining knowledge, making decisions, making predictions, from a set of data [3]. It enables the choosing of the hyper plane space such a way that it closely represents the underlying function in the target space in statistical learning theory the problem of supervised learning is formulated as follows. We are given a set of training data $\{(x1, y1) \ldots (x1, y1)\}$. In $R^n \times R$ sampled according to unknown probability distribution, $P\ (x, y)$, and a loss function, $V\ (y, f\ (x))$ that measures the error, for a given $x, f\ (x)$ is predicted rather than the actual value y. Finding a function f that minimizes the expectation of the error on new data is the problem. Equation 1 shows the function f which minimizes the expected error:

$$\int V\left(y, f(x)\right) P(x, y) dx\, dy \ldots (1)$$

In statistical modeling there should be defined a model of the hypothesis space, which is closest (with respect to some error measure) to the underlying purpose of the target space [8].

1.2 Theory of Learning and Generalization

Early machine learning algorithms aimed to learn representations of elementary functions. Hence, the goal of learning was to output a hypothesis that performed the correct classification of the training data and early learning algorithms were designed to detect such an accurate fit to the data [4]. The ability of a theory to correctly classify data not in the training set is known as its generalization. Another thing to observe is to find where making the best trade-off in trading complexity with the number of epochs; SVM performs better in term of not over generalization when the neural networks might end up over generalizing easily [5]. the illustration as shown in figure 3 brings to light more information about this.

Figure 3: Number of Epochs Vs Complexity

The paper continues to exist as follows: Section 2 deals with an overview of the related research regarding the SVM and its applications, and Section 3 gives comparison of SVM with different technologies whereas the conclusions are drawn in Section 4.

1. SVM: Monolithic Predictor

SVM has been found to be successful when used for pattern classification problems. Using the Support Vector approach to a particular practical problem involves resolving a number of questions based on the problem definition and the design involved with it. One of the major challenges is that of selecting an appropriate kernel for the given application [5]. There are standard choices such as a Gaussian or polynomial kernel, that would be default options, except these prove ineffective or if the inputs are discrete structures more elaborated kernels will be needed. By defining a feature space implicitly, the kernel provides the description language used by the machine for viewing the data. Once the choice of the kernel and its optimization criterion has been made the key components of the system are in place [5]. Let's look at some simple examples.

The task of text categorization is the classification of natural text documents into a specified number of predefined categories based on their content. So that a document can be assigned to more than one category this is not a multi-class classification problem, which can be viewed as a series of binary classification problems, where one for each category. There is a standard representation of text for the purposes of information retrieval provides an ideal feature mapping for constructing a Mercer kernel [6]. Indeed, the kernels somehow incorporate a similarity measure between instances. It is reasonable to assume that experts working in the specific application domain have already identified valid similarity measures in areas such as information retrieval and generative models [6], [7].

Traditional classification approaches perform poorly when working directly because of the high dimensionality of the data, since Support Vector Machines can avoid the pitfalls of very high dimensional representations [12]. A very similar approach to the techniques described for text categorization can also be used for the task of image classification, as in that case linear hard margin machines are frequently able to generalize [5]. The first real-world task on which Support Vector Machines were tested was the problem of handwritten character recognition; and then, multi-class SVMs have been tested on these data. The most interesting thing about SVM is not only to compare SVMs with other classifiers, but also different SVMs compare amongst themselves [14]. They turn out to have approximately the same performance, and continues to share most of their support vectors, independently of the chosen kernel. SVM can perform well as these systems without including any detailed prior knowledge is certainly is a remarkable fact [6].

2.1 SVM for Classification

Even though it's considered that Neural Networks are easier to use for data classification, SVM is a useful technique. However, sometimes unsatisfactory results are obtained. A classification procedure usually done with training and testing data which consist of some data instances [13]. Each instance in the training set contains one target values and several attributes. SVM has to produce a model which predicts the target value of data instances in the testing set which are given only the attributes [5]. Classification in SVM is an example of Supervised Learning. This information points to a desired response, validating the accuracy of the system, and be used to help the system to learn and act correctly. In SVM classification involves identification which is intimately connected to the feature selection or feature extraction. They can be used to identify key sets which are involved in processes distinguish the classes. Feature selection and SVM classification together have a use even when the prediction of unknown samples is not necessary [8].

2.2 SVM for Regression

SVMs can also be applied to regression problems with the introduction of an alternative loss function [5]. The loss function must be modified to include a distance measure. The regression can be linear and nonlinear. Linear models mainly consist of the loss functions such as e-intensive loss functions, quadratic and Huber loss function. Similar to classification problems, a nonlinear model is usually required to adequately model data. As the non-linear approach of SVC. A non-linear mapping can be used to map the data into a high dimensional feature space where linear regression is performed. In the regression method there are considerations based on prior knowledge of the problem and the distribution of the noise. The kernel approach is again employed to address the curse of dimensionality [16].

2. Comparison, Performance and Training

The methods considered are k- Nearest Neighbor (k-NN), support vector machines (SVM), smallest vertex polygon (SMP), neural networks (NN), Bayes theorem (BT), and Markov Chains (MC). A common trait most fingerprinting methods share is that a small change in the layout of the environment or the position of emitter devices would require retraining the system. The NN and MC methods have a similar behavior are discussed together.

3.1 Performance Based Comparision

Some of the most used methods for fingerprinting are k-Nearest Neighbor, Support Vector Machine, Smallest M-vertex polygon, and Neural Networks Apply pattern recognition algorithms over a set of signals (generally RSSI) to determine the current position of a resource. In Table 1 their performance is compared.

Table 1: Comparison of fingerprinting positioning methods in indoor environments

Methods	*Accuracy*	*Precision*	*Scalability*	*Complexity*	*Overall Cost*
kNN	High	High	Medium	High	High
SVM	High	Medium	High	Medium	Medium
SMP	High	Medium	Medium	High	High
NN	High	High	High	Medium	Medium
BT	High	High	Medium	High	High
MC	High	High	Medium	High	High

3.1.1 Performance of KNN on Indoor Environments

The k-NN positioning method has a great accuracy at close range (2.4m at 50m, 1.26 at 25m), but it quickly deteriorates when closing in to the target. This is due to an innate problem of the k-NN algorithm: similar readings (i.e. Close points) increase the probability of an estimation error. Even though the k-NN algorithm doesn't always compute position in the same way, it has a remarkable precision. A problem with k-NN is that greater granularity (more fingerprints) increases the computational needs and requires a greater training effort [16].

3.1.2 Performance of SVM on Indoor Environments

The SVM method has a high accuracy rate although its precision can be affected by similar readings of signals coming from different points. However, the complexity of the operations required for the positioning estimations demands a powerful computing infrastructure. An advantage of this method is its scalability; it is able to support a large amount of simultaneous targets and can be easily adapted to position resources in 3-D environments, because of its multi-dimensional approach [16].

3.1.3 Performance of SMP on Indoor Environments

SMP calculates target positions with the help of averages, which leads to a relatively high accuracy in most cases, but a high precision error rate [16]. This impacts SMP's score in both scalability and complexity, as with other fingerprinting methods.

3.1.4 Performance of NN on Indoor Environments

NN has great accuracy at close range as 2.94m at 50m and 1.39m at 25m [16]. A strong point of NN is that they have better performance than other methods when the training database is very large, though it still requires a moderate amount of training and computing power to carry out acceptable estimations.

3.1.5 Performance of BT and MC on Indoor Environments

The BT and MC fingerprinting have greater accuracy at greater distances from the reference points, which decreases at closer distance [15]. But both methods work under probability assumptions and their complexity is relative to the size of the coverage area and amount of targets. A "re-sampling" can be done at any time for the BT method, allowing the users to adjust marginal distributions of access points when a change invalidates the current signal calibration. A drawback of the MC method is the enormous size of its state space, which grows with each new state update [15].

Sampled SVM is a new data modeling method for training SVMs. Where its shown that it is fast, scalable, and parallelizable, yielding approximate solutions to SVM training problems. Furthermore, the method may use any SVM training implementation. Comparing the eight-core implementations of Sampled SVM and Cascade SVM demonstrates the algorithm to be faster across all four data sets. Additionally, training SVM sub problems in Sampled SVM is highly parallelizable. The main advantage of Sampled SVM over the Cascade SVM is the increase in the amount of work that is parallelizable. This advantage is particularly acute when the ratio of support vectors to dataset size is large.

3.2 Training Procedure

The Cascade SVM divides the global SVM problem into a number of local problems, and then aggregates the results into a global solution as shown in figure 4 [9]. For this purpose The SVM model should be trained by data set.

1. Set up the training data
2. Set up SVM's parameters
   a. *Type of SVM.*
   b. *Type of SVM kernel.*
   c. *Termination criteria of the algorithm.*
3. Train the SVM
4. Regions classified by the SVM
5. Support vectors

For better performance of the system we should define or choose a kernel function carefully. There is a well known method for choosing a kernel function given below [10].

- Transform the data to the format of SVM packages
- Conduct simple scaling of the data
- Consider the Radial Base Function (RBF) kernel $K(x, y) = e^{-\gamma\|x-y\|^2}$ … (2)
- Use cross-validation technique to find the best parameters C and γ

• Use the best parameters C and γ to train the whole training data set
• Test

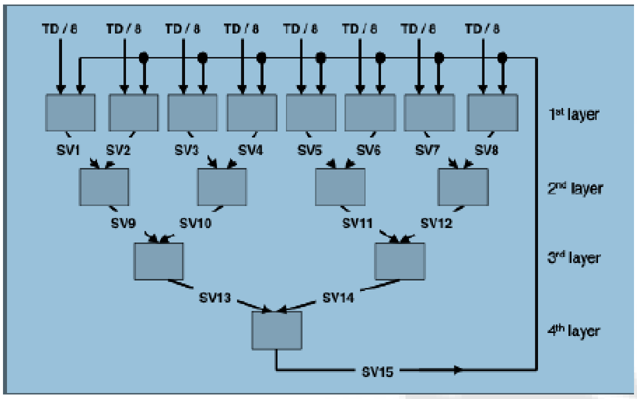

Figure 4: The Cascade SVM divides the global SVM problem into a number of local problems, and then aggregates the results into a global solution.

3. Conclusion

Support Vector Machine, k-Nearest Neighbor, Neural Networks, Decision Tree, and many others are Supervised learning algorithms which are used to solve supervised learning problem. But what makes SVM interesting is the ability to classify complex problems having linearly non separable cases with nonlinear decision boundary and its ability to find the optimum nonlinear decision boundary. Using SVM system model will not get any arbitrary nonlinear decision boundary to separate the classes in the training data set. In fact, SVM algorithm will give the optimum linear separation in high dimensional feature space that will be the equivalent to the optimum non-linear decision boundary in the original data set. SVM has some major features such as convexity, duality, kernels, and sparseness used in Machine Learning. The Support Vector Machine algorithm displays a very low error rate when used for regression (spatial localization). Performance of SVM is fully depends on Kernel functions. In future a technique for choosing and developing a Kernel Function and monolithic control system for predicating data from small training data sets has great scope.

References

[1] Vapnik, V.N., "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on* , Vol. 10, no. 5, pp. 988,999, Sep 1999
[2] OpenCV, "Introduction to Support Vector Machines," [Online]Available: http://docs.opencv.org/doc/tutorials/ml /introduction_to_svm/introduction_to_svm.html? highlight=svm. [Accessed: Nov. 9, 2013].
[3] Theodoros Evgenuiu and Massimilliano Pontil, Statistical Learning Theory: a Primer 1998.
[4] Nello, Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.
[5] Tom Mitchell, Machine Learning, McGraw-Hill Computer science series, 1997.
[6] Osuna E., Freund R., and Girosi F., "Support Vector Machines: Training and Applications", A.I. Memo No. 1602, Artificial Intelligence Laboratory, MIT, 1997.
[7] Veropoulos K., Cristianini N., and Campbell C., "The Application of Support Vector Machines to Medical Decision Support: A Case Study", ACAI99
[8] Olivier Bousquet, Stephane Boucheron, and Gabor Lugosi, "Introduction to Statistical Learning Theory".
[9] Ferragut, E.M.; Laska, J., "Randomized Sampling for Large Data Applications of SVM," *Machine Learning and Applications (ICMLA), 2012 11th International Conference on* , vol. 1, no., pp.350,355, 12-15 Dec. 2012
[10] Hsu, Chih-Wei and Chang, Chih-Chung and Lin, Chih-Jen. "A Practical Guide to Support Vector Classification," Department of Computer Science, National Taiwan University. Year 2003.
[11] Trafalis T., "Primal-dual optimization methods in neural networks and support vector machines training", ACAI99.
[12] Wassi, G. I.; Despins, C.; Grenier, D.; Nerguizian, C., "Indoor location using received signal strength of IEEE 802.11b access point," *Electrical and Computer Engineering, 2005. Canadian Conference on* , vol., no., pp. 1367,1370, 1-4 May 2005.
[13] Duda R. and Hart P., "Pattern Classification and Scene Analysis", Wiley, New York 1973.
[14] M. O. Stitson and J. A. E. Weston. Implementation issues of support vector machines. Technical Report CSD-TR-96-18, Computational Intelligence Group, Royal Holloway, University of London, 1996.
[15] Burgard, Derr, Fox, Cremers, "Integrating global position estimation and position tracking for mobile robots: the dynamic Markov localization approach," *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol.2, pp.730-735.
[16] Wassi, Despins, Grenier, Nerguizian, "Indoor location using received signal strength of IEEE 802.11 b access point." *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1367-1370.

Author Profile

Mr. Abhijit Adhikari received M.E. In Software Engineering from the Computer Science Department of Birla Institute of Technology, Mesra, Ranchi.(2013). Currently he is working as an Assistant Professor in the Department of Computer Engineering, in RMD Sinhgad Technical Institutes Campus, Warje, Pune. He has got a GATE Scholarship by AICTE for P.G. (2011-2013). He has published and presented his research in International Conference on Advanced Computing Methodologies-2013" (ICACM-13). His area of interest is Soft Computing and Data Mining.

Mr. Chetan Nimba Aher recived B.E. In Computer Engineering from Computer Department of S.S.V.P.S. C.O.E., Dhule from North Maharashtra University, Jalgaon. (2012). Currently he is pursuing M.E. In Computer Engineering from RMD Sinhgad School of Engineering, Warje, Pune. He is also working as an Assistant Professor in the Department of Information Technology in AISSMS IOIT, Shivajinagar, Pune. His area of interest is Soft Computing and Computer Networking.