MULTICLASS CLASSIFICATION USING SUPPORT VECTOR MACHINE (SVM)

Ambarish Parthasarathy[1]
ai23resch01001

Pendota Amulya[1]
ee21mtech12003

[1]Indian Institute of Technology, Hyderabad, India

ABSTRACT

Support Vector Machine (SVM) is a popular machine learning algorithm used for both regression and classification tasks. SVMs are particularly useful for classification tasks. In vanilla/basic SVM, the objective is to find the hyperplane that best separates the classes in a way that maximises the margin between the hyperplane and the data points, assuming the data is linearly separable and has only 2 classes. The basic idea of SVM is to find the hyperplane that maximises the margin between the different classes. By maximising the margin, SVM tries to create a decision boundary that is robust to noise and can generalise well to new data points. However, in practical/real-world applications, the data could be more than 2 classes .

INTRODUCTION TO SUPPORT VECTOR MACHINE (SVM)

Machine Learning is one of the key areas, where the application can be used to mimic the human notions of problem solving. One of such areas is the field of Classification and Regression. In order to predict or classify a particular class of a dataset, the simplest approach that can be taken is to sketch the cluster of that individual class of the dataset. The sketched clusters can be separated by a set of straight lines or hyperplanes when referring to higher dimensional analysis. This is the main stage where a Support Vector Machine (SVM) plays its role.

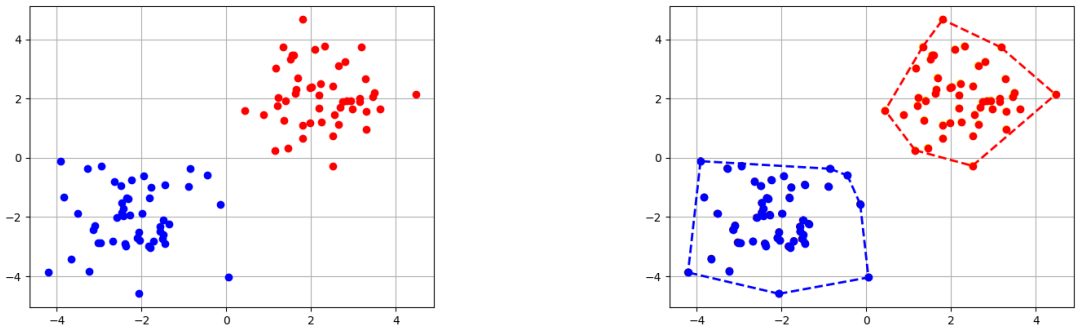


Figure 1: The set of points of two classes, one with the blue and the other with red. It is to be noted that there exists a good separation between the two scatter data (left). The corresponding convex clusters of the respective classes (right). This is also referred to as the convex hull of the two class data. One of the better solutions is to find the minimum distance

between the two convex hulls which in other words is equivalent to maximising the distance between the two scatter points.

Clearly, multiple hyperplanes can be constructed that can separate one of the classes from the another, but rather importantly the hyperplane that is to be constructed, should have maximum margin. This would rather be helpful in scenarios where newer data is injected into the particular model.

In this particular report, the analysis is first done using the two classes which is also referred to as Binary SVM. Further it is also assumed that in the case of the Binary SVM, the data points are linearly separable. In the forthcoming stages, the approach is extended to Multi-class SVM, such as, One to One Method (OvO), One to All Method (OvA), where a more practical approach of Non-linearly separable dataset is worked upon. Then another methodology for classification using the Hinge Losses is analysed. Initially, the analysis is done using the simulation data (using the Gaussian random ets of points in the dataset) and further this analysis is done using the practical IRIS and the MNIST Datasets.

The penultimate section includes the results that were obtained by the above methods and the future scopes that one can work upon in continuation to the analysis of Multi-Class SVM.

BINARY SVM

The first case of classification is the Binary SVM, in which a single hyperplane separates the linearly separable two classed scatterd data, in such a way that the margin between the two classes is maximum. This can be formulated using the following equation:

Considering w_i as the weight vector of size *(d x 1)* and a bias b as a scalar quantity and X as the data points, the objective function is given by:

$$f(x) = arg\ min\ \frac{1}{2}||w||^2$$

$$Subject\ to: y_i(w^T X + b) \geq 1$$

Where the norm of the weight vector is to be minimised and the '1' here represents the margin, meaning the maximum margin is constrained and should be beyond one unit. This expression is valid if and only if the dataset that is being considered is perfectly separable. The corresponding result of the above convex problem is shown in the Figure 2.

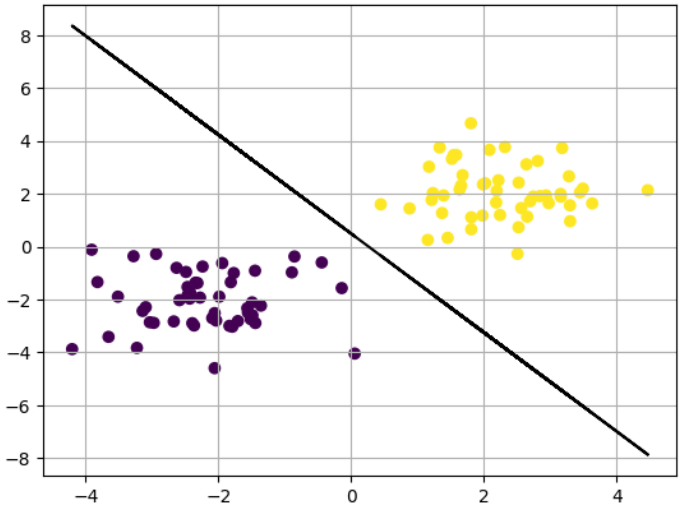


Figure 2: The hyperplane that separates the two scatter data with the maximum margin. This could also be solved by finding the minimum distance between the convex hull of the respective convex hull[1].

In a practical scenario the perfectly separable data isn't always found and importantly, there aren't always the case where there would be only two classes, similar to that of Figure 2. The Binary SVM on the other hand can be extended to multiple classes and also can handle the case of Non-Separable Data which is referred to as Multi Class SVM.

MULTI-CLASS SVM

In scenarios where the data is distributed into multiple clusters corresponding to different classes, a variation of Binary SVM can be used [2]. This method involves drawing multiple hyperplanes to classify the scattered data clusters. A some amount of tolerance is given to the particular model so that it can accommodate some misclassifications. This method is also referred to as Soft Thresholding.

The equation of such a Soft Thresholded Model gets chanced to the following expression :

$$f(x) = arg\ min\ ||w||^2 + \left(\frac{1}{C}\right)\sum(1 - y_i(w^T X + b))$$

$$Subject\ to : y_i(w^T X + b) \geq 1 - \zeta$$

Where, ζ represents the small tolerance given to the particular model. Higher this tolerance, higher is the misclassification that is allowed.

[1] https://www.youtube.com/watch?v=NT7C228pzOA&list=PLMmTBP3nE-rROSIYO419Bp_dZNpjWeoRG&index=24

Approaches for Multi-Class Classification

1. One Versus One Approach (OvO)

The One Versus One Approach is one of the methods of solving the Multi-Class classification problem . This method uses present class data along with all the other class data present in dataset pairwise, i.e two classes at a time and constructs a hyperplane corresponding to it. If there are N classes present in the dataset then N(N-1)/2 no of binary SVM are required for a one vs one approach [4,5].

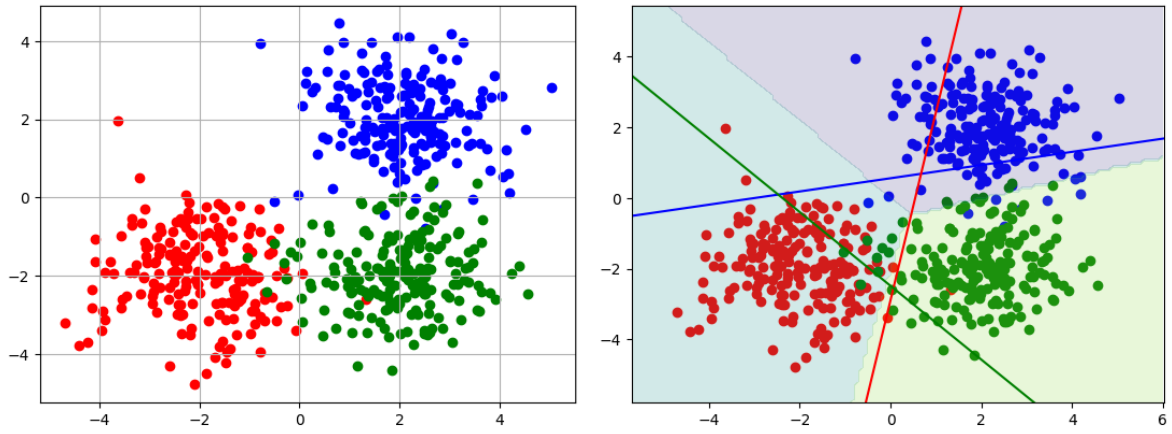


Figure 3 : The multiclass data (left) and the corresponding hyperplane separation using OvO approach(right).

2. One Versus All Approach (OvA)

The One Versus All (OvA) Approach on the other hand, takes the present data and compares it relative to that of all the other data present in the model. This gives a better understanding of the behaviour of each of the data present in the model. If there are N classes present in the dataset, the N binary SVM classifiers are required to find the optimal hyperplanes in one vs all approach [4,5].

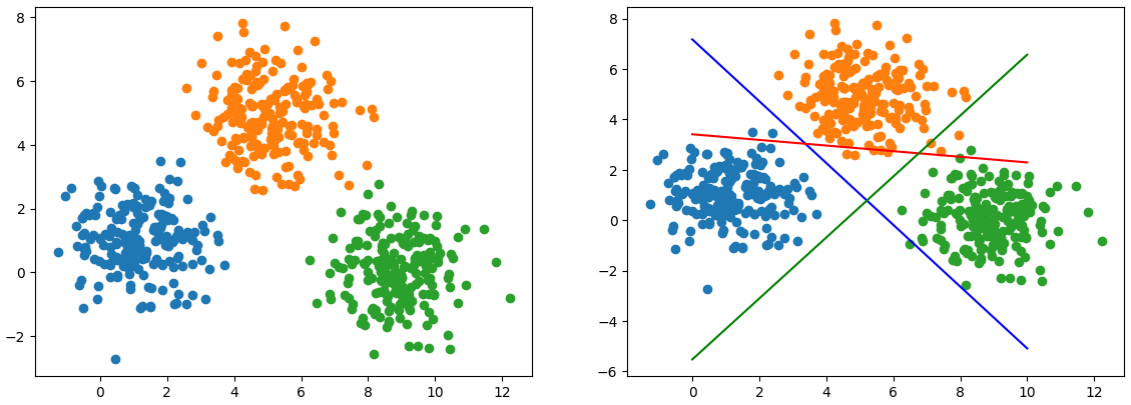


Figure 4: The One Versus All Approach (OvA) applied to a 3-class data.

Multi-Class CLassification using Hinge Loss Function

The motivation for using hinge loss in multi-class SVM is to encourage the algorithm to find a large margin classifier that can generalise well to new data points, by penalising the SVM for incorrect classifications and encouraging correct classifications with a large margin.. The greater the distance between a data point and the decision boundary, the smaller the loss. The algorithm extends to multi-class classification by choosing the class with the highest score among all the SVMs.

In multi-class classification using hinge loss, we consider a dataset of D-dimensional input data samples with *N* classes and *M* samples. We can represent the score equation:

$$f(x): W * X + b$$

$$y_{pred} = arg\ max_i(f(x))$$

where *W* is a matrix of *NxD* with each row corresponding to optimal parameters of the respective class, *X* is an *MxD* matrix with M training samples each *D* dimension, and *b* is an *Nx1* bias vector. To avoid using a separate bias term, we include it in *W* and *X*, resulting in an updated weight matrix W of size *Nx(D+1)* and updated *X* of size *Mx(D+1)*. Y_i is the label which corresponds to samples in X_i.

When we perform $X * W^T$ the score matrix ($f(x)$) is an *MxN* order matrix, where each element represents the probability that the i^{th} sample belongs to the k^{th} class (say a_{ik}) . The SVM algorithm with hinge loss function ensures that there is a sufficient gap between the score for the correct class and the scores for all other classes. In other words, if the i^{th} sample belongs to the k^{th} class,

then $(a_{ik} - a_{ij}) > \varepsilon$, where j!=k and j=1,2...,10,

This condition can be captured using the hinge loss function, which is

$$L_i = max(0, a_{ij} - a_{ik} + \varepsilon)$$

On further addition of regularisation term to the loss function in order to penalise large weights of *W* matrix which can lead to overfitting, the total loss equation becomes

$$L = \sum_i^M \sum_{j \neq k}^N max(0, w_j x_i - w_k x_i + \varepsilon) + \lambda ||w||_F^2$$

Where $a_{ij} = w_j x_i$ and $a_{ik} = w_k x_i$

The squared Frobenius norm is considered which discourages big weights and so provides a preference for a certain set of weights over others. This reduces the ambiguity and also boosts generalisation performance. λ ,here is a hyperparameter, which weights the regularisation penalty.

The optimal weight matrix can be calculated using the gradient descent approach with the help of gradients. The standard gradient descent equation for updation would be

$$x_{i+1} = x_i - \delta \nabla f(x_i)$$

Differentiating L with respect to w_k and w_j, we get

$$L_{wk}' = -\frac{1}{M}\sum_i \sum_{j \neq k} [Xi \,|(w_j^T X_i - w_k^T X_i + \varepsilon > 0)] + \eta w_k$$

$$L_{wj}' = \frac{1}{M}\sum_i \sum_{j \neq k} [Xi \,|(w_j^T X_i - w_k^T X_i + \varepsilon > 0)] + \xi w_k$$

Substituting the above equations in gradient descent equation, we get

$$W_k(p+1) = Wk(p) + \frac{1}{M}\sum_i \sum_{j \neq k} [Xi\,(w_j^T X_i - w_k^T X_i + \varepsilon > 0)] - \eta w_k(p)$$

$$= (1 - \eta)Wk(p) + \frac{1}{M}\sum_i \sum_{j \neq k} [Xi\,(w_j^T X_i - w_k^T X_i + \varepsilon > 0)]$$

Similarly, we get W_j where $j \neq k \; \forall \; 1 < j < M$

$$W_j(p+1) = W_j(p) - \frac{1}{M}\sum_i \sum_{j \neq k} [X_i \,|(W_j^T X_i - W_k^T X_i + \varepsilon > 0)] - \xi W_j(p)$$

$$= (1-\xi) W_j(p) - \frac{1}{M}\sum_i \sum_{j \neq k} [X_i \,|(W_j^T X_i - W_k^T X_i + \varepsilon > 0)]$$

EXPERIMENTAL RESULTS

We have experimented with the IRIS dataset which contains 150 data points with 3 labels. The dataset is split in a 70:30 configuration meaning 70% corresponds to the train set and 30% of the data corresponds to the test set.

Table 1 shows the accuracy and the loss calculated over the IRIS test dataset along with the confusion matrix.

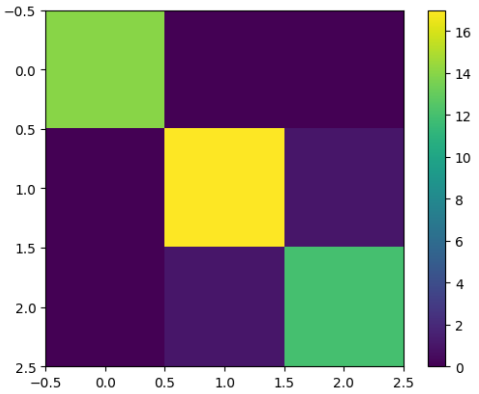
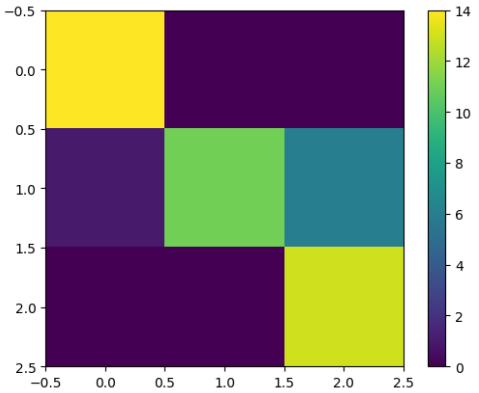
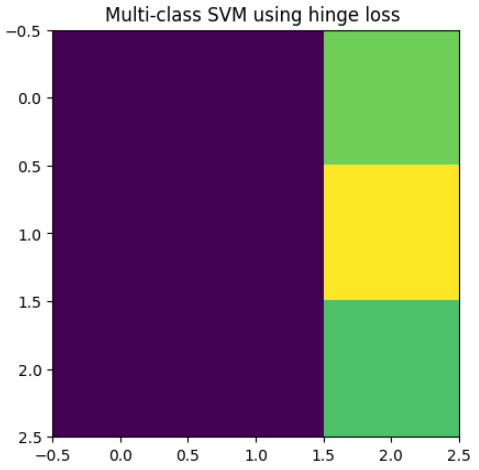
	Accuracy	Loss/error	Confusion matrix
One vs One	95.5	4.5	
One vs All	84.4	15.6	
Using Hinge loss function	-	-	

Table1: The Multi-Class SVM Results along with the Accuracy, Loss and the confusion matrix.(all the experimental codes are available here)

The failure of obtaining undesired confusion matrix in case 3 is discussed in the upcoming section.

We have also experimented Multi-class classification using hinge loss function algorithm over MNIST dataset which has 10 classes with 60,000 training images and 10,000 test

images. For computational and time constraint we have considered 500 samples for training and 200 samples for testing the algorithm. This resulted with a test accuracy of 92 percent and a test error of 8 percent for 200 test samples

DRAWBACKS OF THE DISCUSSED METHODS

The initial multi class classification experiments with random Gaussian iid samples shows the drawback of one vs one and one vs all approaches.

- If a test point falls in the region that is in a common or intersection region of 2 classes, the hyperplane might not accurately differentiate and conclude to which class the point belongs to.
- If the test point falls in the region that doesn't specify any class, then using the optimal hyperplanes obtained will not classify that particular point.for example a point inside the triangular region.
- Class imbalance is another important drawback to notice in one vs all approach while training. This is because we consider one class with say N samples against (M-1) classes with a total of (M-1)*N samples which creates an imbalanced dataset.
- If the number of classes increases then the number of binary SVMs required in one vs one and one vs rest algorithms increases which leads to huge computational cost.
- The complicated part while using a hinge loss for multiclass classification would be in deciding the hyperparameters. The selection of hyperparameters like λ, ξ, η, ε and the no of iterations would play a key role in the accuracy and the loss that we obtain. This could be one of for the undesired confusion matrix that was obtained which was show in Table1 above

FUTURE SCOPE

The following points listed below can be worked upon as a future task:

- The performance of the Multiclass classification using hinge loss algorithm can be improved further with appropriate selection of hyperparameters.
- Several other methods like the DAG-SVM where DAG-SVM organises the classes into a directed acyclic graph (DAG) structure, where each node corresponds to a class and the edges signify the hierarchical connections between the classes. The approach builds a number of binary SVM classifiers to divide the classes into groups based on the DAG topology and another area to explore [4].
- Improved versions of DAG -SVM can be the next focused methods for further evaluating this problem
- We can also extend the above discussed methods using kernel tricks i.e projecting the data to higher dimension and then classifying it using different kernels.

CONCLUSION

The basic idea of binary SVM is to find the hyperplane that maximises the margin between the different classes. By maximising the margin, SVM tries to create a decision boundary that is robust to noise and can generalise well to new data points.

However, in practical or real-world applications, the data could be more than 2 classes and hence in this draft we tried to focus on analysing the existing methods and their drawbacks with the help of concepts learnt during convex optimization courses. In this draft, we have analysed how a simple binary classifier SVM could be extended for multi-class classification. We have done an experimental analysis of the existing methods, which include one vs one and one vs all and multi-classification using hinge loss, and the results are shown on standard datasets i.e, IRIS and MNIST datasets.

REFERENCES

[1] Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. No. 4. New York: springer, 2006
[2]Stanford CS Lecture CS231n: Convolutional Neural Networks for Visual Recognition
[3] Liu Y, Zheng YF. One-against-all multi-class SVM classification using reliability measures. InProceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. 2005 Jul 31 (Vol. 2, pp. 849-854). IEEE.
[4]Sabzekar M, GhasemiGol M, Naghibzadeh M, Yazdi HS. Improved DAG SVM: A New Method for Multi-Class SVM Classification. InIC-AI 2009 Aug 14 (pp. 548-553).
[5]C.-W. Hsu and C.-J. Lin, “A comparison of methods for multi-class support vector machines,” IEEE Transactions on Neural Networks, vol. 13, pp. 415–425, 2002, doi: 10.1.1.19.4258.