SMO implementation from Scratch and application of SVM on a good dataset

1st Addanki Veerababu
dept Of Computer Science (Artificial Intelligence)
School Of Engineering,Amrita Vishwa VidyaPeetham
Clappana PO ,Kollam,Kerala,India

2nd Dhanush Krishna R
dept Of Computer Science (Artificial Intelligence)
School Of Engineering,Amrita Vishwa VidyaPeetham
Clappana PO ,Kollam,Kerala,India

3rd Durgapu Sathvik
dept Of Computer Science (Artificial Intelligence)
School Of Engineering,Amrita Vishwa VidyaPeetham
Clappana PO ,Kollam,Kerala,India

4th Kolla Dorasanaiah
dept Of Computer Science (Artificial Intelligence)
School Of Engineering,Amrita Vishwa VidyaPeetham
Clappana PO ,Kollam,Kerala,India

5th Pranav Jayasankar Nair
dept Of Computer Science (Artificial Intelligence)
School Of Engineering,Amrita Vishwa VidyaPeetham
Clappana PO ,Kollam,Kerala,India

***Abstract*—In this paper, Support vector Machines(SVM) is implemented on a MINST dataset and Sequential minimal optimization (SMO) is implemented from scratch.SMO is a optimization technique which is used to break this large quadratic programming(QP) into a series of smallest QP problems. SMO consumes linear amount of memory in the training set size and also handles very large amount of training sets.Because calculating of matrix is avoided, SMO requires between linear and Quadratic in training sample,while chunking algorithm requires between linear and cubic in training samples.SVM is implemented using in-built libraries in python on a MINST dataset.**

***Index Terms*—Support Vector Machine,Sequential minimal optimization, MNIST**

I. Introduction

Training of SVM algorithm is slow for large data-sets and it is complex to implement.John C. Platt from Microsoft in 1998 invented a new fast algorithm for training support vector machines known as Sequential minimal optimization(SMO).Because of its peculiar features like low Computation memory and very fast computation it is so popular and used in various sectors for training SVM.It is more faster than chunking algorithm which is used widely before SMO.

A. Support Vector Machines

Support Vector Machine is a supervised machine learning algorithm that can be used for both classification and regression problems but it is used mostly in classification.The main In SVM want to find the optimal hyperplane that maximizes distance between closest points(support vector machines) and hyperplane. In SVM ,there are two types of margin classifications.first one is Hard margin which doesn't allow any training samples into margin areas.next one is Soft margin which allows some samples into margin areas which know as miss classifications .For linear separable data we can easily segregates different classes,but for non-separable data required different types of kernels like Quadratic ,linear, Gaussian which discussed later in paper which convert lower dimensional data to high dimensional data.

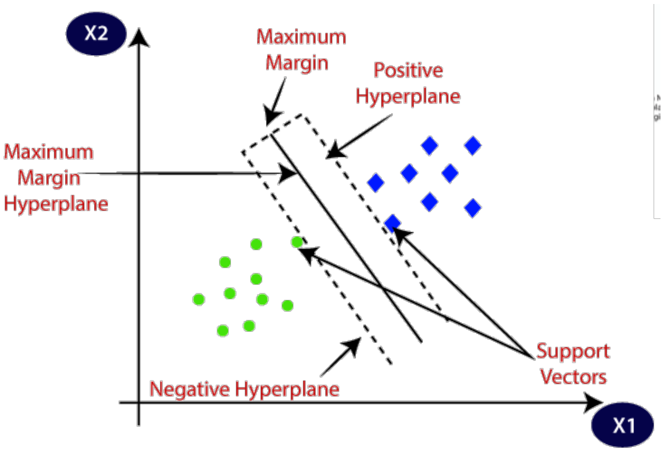


Fig. 1.

B. Sequential minimal optimization(SMO)

Sequential minimal optimization is a iterative algorithm that solved the SVM Quadratic programming(QP) problem.SMO divides the QP problem into small QP sub problems.It doesn't take any matrix memory for computation.The main objective of SVM is finding two Lagrange multipliers.If we one Lagrange multiplier it didn't satisfies the linear equality constraint condition.The dual optimization function in α is

$$\min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j (\vec{x}_i \cdot \vec{x}_j)\alpha_i \alpha_j - \sum_{i=1}^{N}\alpha_i$$

where,subject to inequality constraint

$$\alpha_i \geq 0, \forall i$$

and subject to equality constraint

$$\sum_{i=1}^{N} y_i \alpha_i = 0$$

There are mainly two steps in Sequential minimal optimization.Step one is finding the two Lagrange multipliers and second one is choosing the next two multipliers for optimizing.For choosing the two α for optimizing ,they should violates the below KKT conditions ,

$$\alpha_i = 0 \Leftrightarrow y_i u_i \geq 1$$
$$0 < \alpha_i < C \Leftrightarrow y_i u_i = 1$$
$$\alpha_i = C \Leftrightarrow y_i u_i \leq 1$$

For non-separable data,the The dual optimization function in α is

$$\min_{\vec{w},b,\vec{\xi}} \tfrac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i \text{ subject to } y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1 - \xi_i, \forall i$$

For non-separable data the Lagrange multiplier(α) should not greater than C.

$$0 \leq \alpha_i \leq C, \forall i$$

C. Heuristics for Multipliers to choose and optimize

After each step, SMO alters and optimizes 2 Lagrange Multipliers of which atleast one of them will violate the KKT conditions before the step. Therefore, as per Osuna's theorem, the objective function of each step will decrease. Convergence is thus guaranteed. Heuristics are used in SMO to determine which of the 2 Lagrange multipliers need to be jointly optimized. SMO uses 2 choice heuristics.

The first-choice heuristic provides the outer loop of the SMO algorithm This outer loop is used to iterate over the training set. If an example in the set violates the KKT conditions, it is qualified for optimization. After the first pass, the loop iterates over those examples in which the Lagrange multipliers are neither C nor 0. Once again, these are checked against the KKT conditions to determine eligibility for optimization. Next, the outer loop passes over non-bound examples repeatedly till the KKT conditions are met. This process is repeated until the entire train set meets the KKT conditions. In this method, the non-bound subset, i.e., those sets that are more likely to violate KKT. The SMO algorithm will thus iterate over the non-bound subset until that subset is self-consistent, then SMO will scan the entire data set to search for any bound examples that have become KKT violated due to optimizing the non-bound subset.

After obtaining the first Lagrange multiplier, the second multiplier is chosen to maximize the size of the joint optimization step. As finding the kernel function (K) is time consuming, SMO is used to approximate the step size by the absolute value of the numerator of the equation —E1-E2—. We keep a cached error value E for every non-bound example in the training set and then chooses an error to approximately maximize the step size.

If E1 → positive: SMO chooses example with minimum error E2

If E1 → negative, SMO chooses example with maximum error E2

Under some circumstances positive progress of SMO is not possible using second option heuristic Eg: if 1st and 2nd training examples share identical input vectors x the positive progress cannot be made also the objective function will become semi-definite in that case. Hierarchy of second choice heuristics is used by SMO until it finds a Lagrange multipliers pair which makes a positive progress. Also, by making a non-zero step size upon the joint optimization of 2 Lagrange multipliers the positive progress can be determined. The SMO will start iterating through non-bound examples if the heuristic does not make any positive progress. SMO will start iterating through the entire training set If none of the examples makes a positive until an example is found that makes a positive progress. Both the iteration is started at random locations it makes sure that bias of SMO towards beginning is not possible. In extreme circumstances none of the examples will make adequate second example Then the first example is skipped, and SMO continues with another example which is chosen as the first one.

II. Literature Review

In previous years, large SVM learning problems are optimized into a small tasks. This whole process done by an algorithm known as chunking algorithm. The problem with chunking algorithm is that while scaling with support vectors to solve Quadratic Programming problems. So, an algorithm called Sequential minimal optimization(SMO) is invented by John C Platt in 1998 to deal with the problem in chunking algorithm. When compared with chunking algorithm SMO is more faster about thousand times.The specific version of the Osuna algorithm is Sequential minimal optimization(SMO). The closely related optimization algorithm for Sequential minimal optimization(SMO) is Bergman or row action methods. Libsvm library is one of the famous libraries to implement SMO algorithm.

III. Datasets and Methods

A. Datasets

The data set which is used for the implementation of SMO from scratch and application of SVM is MNIST Dataset. This dataset contains 10000 rows and 785 columns,where each row indicates pixels of digit.Each image consists of 28 pixels in width and 28 pixels in height.Out of 785 columns one column indicates the label which consists of digits ranging 0 to 9.The pixel integers ranging from 0 to 255.

B. Methods

Steps used for solving SMO:

Step 1:

Initially the algorithm initialises the values to the alphas. The Steps 2,3,4 are iterated until convergence condition is met.

Step 2:

From all the available alphas the algorithm exactly chooses two alphas.For the given problem, these alphas represent the possible smallest optimization problem.These alphas are called Lagrange multipliers. This algorithm requires two alphas because it should satisfy the linear equality constraints. By using heuristics the good pair of alphas are chosen.That means at each iteration two alphas are picked from the full vector of alphas and it rate of convergence is increases. For choosing the pair of alphas there are n(n-1) possibilities.The outer loop of the SMO algorithm provided the first Langrange multiplier.The violation of KKT conditions are checks at each sample and it is easy to compute the conditions per sample.It marked the optimization which is required while one sample violates the KKT conditions.

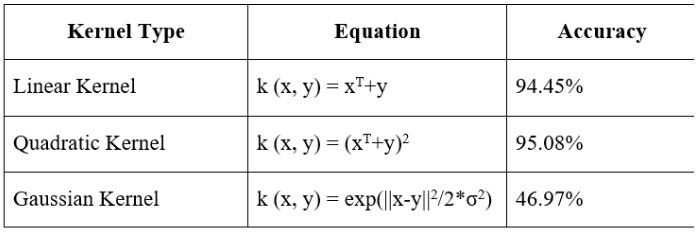
Step 3: In this step, to find the optimal values for multipliers the alphas are optimizes at a time. The other aplhas are remain constant and these two alphas are optimized. In this step there is no use of inner iteration because it represents a closed form solution which is analytically fast to compute. As this algorithm consists many sub QP problems, it is more efficient compared with SVM. In this step another matrix is not required for the storage of full QP problem. To store 2*2 matrix this step needs minor amount of memory at each iteration.

Step 4: Until the algorithm converges, repeat Step two and three. For the SMO algorithm the KKT conditions are used as convergence criteria. All alphas should satisfy this criteria. By satisfying the above criteria the SMO algorithm is converged.

IV. Results and Discussions

A. Implementation of SMO

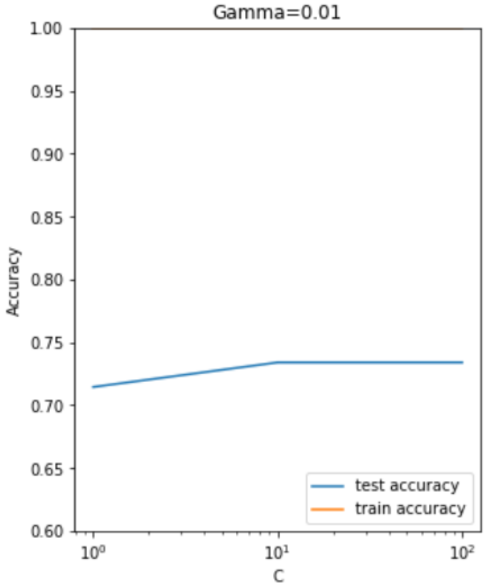
In this paper SMO is implemented from scratch and it is tested on the data set called MNIST. The Kernels are used to convert data from one form to the required form. The linear, quadratic and Gaussian kernels are used and their accuracy's are compared.

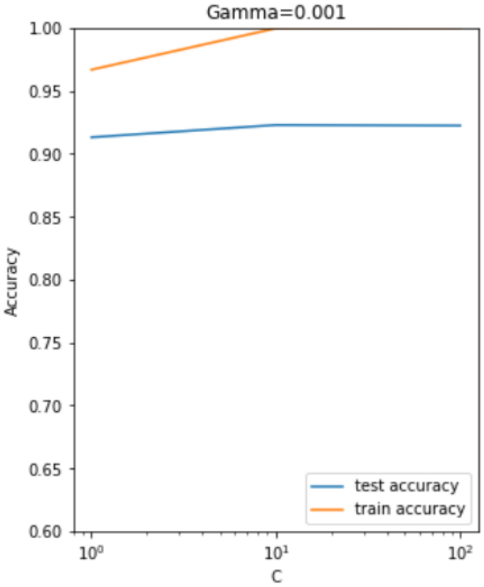


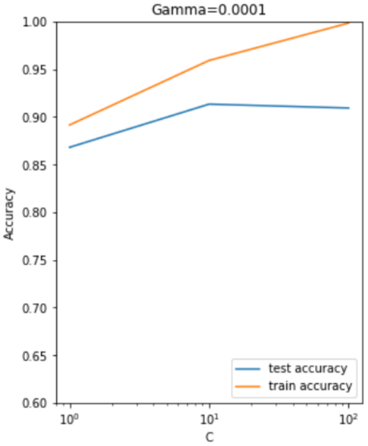
Kernel Type	Equation	Accuracy
Linear Kernel	k (x, y) = x^T+y	94.45%
Quadratic Kernel	k (x, y) = $(x^T+y)^2$	95.08%
Gaussian Kernel	k (x, y) = exp(||x-y||^2/2*σ^2)	46.97%

B. SVM on a good dataset

The SVM algorithm is implemented using MNIST dataset and the following results are obtained for different values of gamma.







The optimal values chosen for this problem are C=1 and gamma=0.001. The final accuracy on the test data is about 94 percent. The accuracy can be increased by using the entire training data.

V. Conclusions

From the experiments conducted above, the SMO algorithm is fast when compared with SVM. Because, SMO soves big QP into a smaller QP problems analytically. The computation time is less for SMO when compared with SVM. Different Kernels are used and their accuracy's are also compared. For the MNIST Data set the Quadratic kernel has more accuracy when compared with linear and Gaussian kernels. SMO training algorithm has the chance to become the standard training SVM algorithm because of its low computation and good scaling with training data.

References

[1] Platt, John. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in Kernel Methods-Support Vector Learning. 208.

[2] https://analyticsindiamag.com/understanding-the-basics-of-svm-with-example-and-python-implementation/.

[3] http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications.

[4] http://yann.lecun.com/exdb/mnist/.

[5] https://www.kaggle.com/c/digit-recognizer.

[6] http://chubakbidpaa.com/svm/2020/12/27/smo-algorithm-simplifed-copy.html.