SLAAG: Efficient SVM Solution to Aggregate Output Learning Problem

Anshuman Dutt **Rahul Kumar**
Computer Science & Automation
Indian Institute of Science, Bangalore

Abstract:

Supervised learning is a classic data mining problem where one wishes to be be able to predict an output value associated with a particular input vector. A new twist on this classic problem was presented by Musicant et al in [1], where, instead of having the training set contain an individual output value for each input vector, the output values in the training set are only given in aggregate over a number of input vectors. This new problem arose from a particular need in learning on mass spectrometry data, but could easily apply to situations when data has been aggregated in order to maintain privacy. Musicant et al examined how support vector machines and other traditional machine learning algorithms can be adapted for this problem. But their approach involving SVM is based on solution of mixed integer quadratic programming which needs impractically large amount of time to give solution. To overcome this we propose here a new formulation which is magnitudes of times faster and also maintains reasonably well accuracy.

1. Introduction:

Supervised learning is a classic data mining problem where one wishes to be able to predict an output value associated with a particular input vector. A predictor is constructed via the use of a training set, which consists of a set of input vectors with a corresponding output value for each. This predictor can then be used to predict output values for future input vectors where the output values are unknown.

Above technique cannot be applied in the analysis of single particle mass spectrometry (SPMS) data [1] where challenge is, we do not have a single output value for each input vector. Instead, each output value in our training set represents the sum of the true output values (which are unknown to us) for a series of input vectors in our training set. We wish to learn a predictor that will still produce individual output values for each new input vector, even though the training set only contains these output values in aggregate. It is important to note that in addition to SPMS, this framework would easily apply to training data which has been aggregated for purposes of privacy preservation.

In [1], Musicant et al introduced a new supervised learning problem, which they refer to as the aggregate output learning problem. They presented a new formal framework for this new machine learning problem and examined how support vector machines and some other traditional machine learning algorithms can be adapted for this problem. In developing an SVM approach for solving this problem they provided a formulation which was based on idea of semi supervised learning problem given in [2]. But this approach involves solution of mixed integer quadratic programming which needs impractically large amount of time to give solution. We will refer to it as MIQP formulation in further sections.

We propose a new algorithm for this problem which is based on another approach for solving semi supervised learning problem given in [3]. In this approach concave minimization problem is solved using successive linear approximation algorithm [5] .Such an approach has been successfully used on a number machine learning, data mining and other problems [3,5,6]. Our experimental study shows that our formulation is magnitudes of times faster in time and also maintains reasonably well accuracy. We will refer to our algorithm as SLAAG (Successive Linear Approximation for AGgregate output problem) in further sections.

We briefly outline the contents of report now. In section 2, we briefly introduce this new problem with the help of an appropriate example. In section 3 we briefly state the SVM formulation given in [1] for aggregate output learning problem. In section 4 we present our new algorithm for this problem. Finally, we present our experiments and comparison with earlier approach in section 5.

2. Problem Formulation:

First, we point out that the traditional classification problem, in general, allows each input vector to belong to one of an arbitrary number of classes. We constrain ourselves to the binary classification problem, where each input vector belongs to one of two possible classes.

Table 1 shows a sample dataset for our framework. Suppose that we are given a training set of input vectors where each has an unknown output value (also known as the class label). This output value is a "yes" or a "no," depending on to which class its corresponding input vector belongs.

The training set is divided into collections of input vectors where aggregate output values are known for each collection. The goal is the same as for the traditional classification problem. We wish to learn a predictor f, using the training set only, that performs well on unseen test data presumably drawn from the same source.

Note that f operates on a single input vector and produces a single output value, though the training set contains output values aggregated over multiple input vectors.

Age	Income	Weight	Beer over Wine?
50	75000	220	
30	56000	180	3 Yes
50	60000	170	1 No
19	2000	150	
32	60000	160	
35	90000	180	1 Yes
60	85000	165	3 No
53	92000	190	

Age	Income	Weight	Beer over Wine?
40	48000	170	?
29	60000	180	?
57	18000	195	?

Table 1 Sample classification training and test sets. In the training set, classifications are known only in aggregate.

3 MIQP formulation:

In developing an SVM approach for solving aggregate output learning problem, it can be observed that the problem is somewhat similar to Semi supervised learning problem [2, 3]. The semi supervised learning problem consists of both labelled and unlabeled training data, and the goal is to use the unlabeled data to improve classification accuracy over using just labeled data. The semi supervised learning problem formulation as given in [2] –

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\eta,\xi,z} & C\left[\sum\limits_{i=1}^{\ell}\eta_i+\sum\limits_{j=\ell+1}^{\ell+k} min(\xi_j,z_j)\right]+\parallel \mathbf{w} \parallel \\ subject\ to & y_i(\mathbf{w}\cdot x_i+b)+\eta_i\geq 1 \quad \eta_i\geq 0 \quad i=1,\ldots,\ell \\ & \mathbf{w}\cdot x_j-b+\xi_j\geq 1 \quad \xi_j\geq 0 \quad j=\ell+1,\ldots,\ell+k \\ & -(\mathbf{w}\cdot x_j-b)+z_j\geq 1 \quad z_j\geq 0 \end{array} \tag{1}$$

Here example x_1 ...x_l are labelled examples and x_{l+1} ...x_{l+k} are unlabeled examples.

This problem is a concave minimization problem as the nonlinear term $\min\{\xi , z\}$ in the objective function is concave because it is the minimum of two linear functions. To solve this Bennett and Demiriz proposed following formulation -

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\eta,\xi,z,d} & C\left[\sum\limits_{i=1}^{\ell}\eta_i+\sum\limits_{j=\ell+1}^{\ell+k}(\xi_j+z_j)\right]+\parallel \mathbf{w} \parallel \\ subject\ to & y_i(\mathbf{w}\cdot x_i-b)+\eta_i\geq 1 \quad \eta_i\geq 0 \quad i=1,\ldots,\ell \\ & \mathbf{w}\cdot x_j-b+\xi_j+M(1-d_j)\geq 1 \quad \xi_j\geq 0 \quad j=\ell+1,\ldots,\ell+k \\ & -(\mathbf{w}\cdot x_j-b)+z_j+Md_j\geq 1 \quad z_j\geq 0 \quad d_j=\{0,1\} \end{array}$$

The idea here was to add a 0 or 1 decision variable, d_j, for each point x_j in the unlabeled set. This variable indicates the class of the point. If $d_j = 1$ then the point is in class 1 and if $d_j = 0$ then the point is in class −1. This results in a mixed integer program.

Based on the above formulation, Musicant et al proposed MIQP formulation by using the fact that for the aggregate output learning problem, we do not know to which class each training point belongs. We do know how many points from each class that there are supposed to be in each aggregate collection, though. Therefore, they proposed the following formulation –

$$\begin{array}{ll} \min\limits_{(\mathbf{w},b,\boldsymbol{\xi}\geq 0,\mathbf{z}\geq 0,\boldsymbol{\eta_c})} & \frac{1}{2}||\mathbf{w}||_2^2+C\sum_i(\xi_i+z_i)+D\sum_c\eta_c \\ \text{s.t.} & \mathbf{w}\cdot\mathbf{x}_i-b+\xi_i+M(1-d_i)\geq 1 \\ & -(\mathbf{w}\cdot\mathbf{x}_i-b)+z_i+Md_i\geq 1 \\ & -\eta_c\leq y_c-\sum_{l\in c}d_l\leq\eta_c \end{array}$$

Here, the subscript c is used to represent an individual aggregate collection; $\ell \in c$ represents the indices of all input vectors associated with collection c, y_c represents the actual number of points associated with class 1 for collection c and η_c is the difference between the predicted and the

actual count of the number of points in class 1 for an aggregate collection c. They therefore sum this error over all points and add it to the objective function, multiplying it by a parameter D to balance the importance of matching the desired aggregate accuracy level for each collection.

This is clearly a mixed integer program with a quadratic objective function, which is known to be slow for even medium sized datasets. To overcome this, making objective function linear would not help much as it is still a mixed integer program. Hence, in the next section we propose an entirely different approach which does not involve mixed integer programming.

4. SLAAG algorithm:

In [3], Mangasarian et al proposed a different approach for Semi Supervised problem given in eq(1), which uses concave minimization procedure and solves it using finite successive linear approximation algorithm. This algorithm is as follows –

(1) Start with a random $(\xi^0, z^0) > 0$

(2) Having (ξ^i, z^i) determine $(\xi^{i+1}, z^{i+1}, w^{i+1}, b^{i+1})$ by solving the following linear program

$$\min_{w,b,\xi,z} \; ||w|| + D\sum_{i=1}^{l}\eta_i + C\,\delta\left(\min(\xi^i, z^i)\right)^T \begin{bmatrix}\xi - \xi^i \\ z - z^i\end{bmatrix}$$

$$\begin{aligned} \textit{subject to} \quad & y_i(\mathbf{w}\cdot x_i + b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1,\ldots,\ell \\ & \mathbf{w}\cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\ & -(\mathbf{w}\cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0 \end{aligned}$$

Where the supergradient $\delta\left(\min(\xi^i, z^i)\right)$ is defined as

$$\delta\left(\min(\xi^i, z^i)\right) = \begin{cases} \begin{pmatrix}1\\0\end{pmatrix} & \text{if } \xi^i < z^i \\ \begin{pmatrix}0.5\\0.5\end{pmatrix} & \text{if } \xi^i = z^i \\ \begin{pmatrix}0\\1\end{pmatrix} & \text{if } \xi^i > z^i \end{cases} \qquad (2)$$

(3) Stop when the value of objective does not change otherwise repeat step 2.

Starting with random (ξ^0, z^0) does not lead us to global optimal answer. In [3] they have suggested another linear program to get this initial values of (ξ^0, z^0) given as follows:

$$\min_{w,b,\xi,z} \; ||w|| + D\sum_{i=1}^{l}\eta_i + \frac{C}{2}\sum_{i=l+1}^{l+k}(\xi_i + z_i)$$

$$\begin{aligned} \textit{subject to} \quad & y_i(\mathbf{w}\cdot x_i + b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1,\ldots,\ell \\ & \mathbf{w}\cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\ & -(\mathbf{w}\cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0 \end{aligned}$$

Now, since our problem does not have any labelled data, hence the given algorithm does not work in its original form. It needs changes to be made at appropriate places. So, in step (1) for finding the initial values of (ξ^0, z^0), we propose the following linear program :

$$\min_{w,b,\xi,z} \frac{1}{2}||w||^2 + C\sum_i a^i.\xi^i + D\sum_i b^i.z^i$$

Subject to *(3)*

$$\begin{aligned} \mathbf{w}\cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\ -(\mathbf{w}\cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0 \end{aligned}$$

Here, a^i and b^i are defined as follows:

$$\mathbf{a^i} = \frac{\textit{the number of examples of class A in collection c}}{\textit{size of collection c}}$$

$$\mathbf{b^i} = \frac{\textit{the number of examples of class B in collection c}}{\textit{size of collection c}}$$

where c denotes the collection in which i^{th} example is present.

We now propose our algorithm for aggregate output learning problem-

SLAAG Algorithm:

(1) Start with (ξ^0, z^0) obtained by solving linear program given above as (3).

(2) Having (ξ^i, z^i) determine $(\xi^{i+1}, z^{i+1}, w^{i+1}, b^{i+1})$ by solving the following linear program

$$\min_{w,b,\xi,z} \frac{1}{2}||w||^2 + E\,\delta(\min(\xi^i, z^i))\begin{bmatrix}\xi - \xi^i \\ z - z^i\end{bmatrix}$$

Subject to

$$\begin{aligned} \mathbf{w}\cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\ -(\mathbf{w}\cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0 \end{aligned}$$

Where the supergradient $\delta(\min(\xi^i, z^i))$ is defined as above in (2).

(3) Stop when accuracy with respect to aggregate information does not increase. Otherwise, repeat step (2).

Hence, we have made appropriate changes to Mangasarian et al. Algorithm [3] in order to solve the aggregate output problem using successive linear approximation procedure. We have used the given aggregate information of the output values to get the starting values of (ξ^0, z^0) and also in stopping criteria. In our experiments we have found that the formulation given in (3) gives very good approximation of the solution and then the step (2) of the above algorithm takes at most 5-7 steps to converge to the final solution.

5. Experiments

For experiments we used same datasets that were used by Musicant et al [1]. First is, *breast-cancer Wisconsin*[4] contains data used to determine whether a tumour is malignant or benign. This set contains 11 features, the last of which is a binary classification of the tumour. We ignore the first feature, a subject identification number since it is not relevant to our analysis. The other features consists of integer values between 1 and 10 that approximate continuous diagnostic measurements. *Dermatology* [4], is our second dataset. This contains 33 features for determining the type of Eryhemato-squamous disease in patients. All values are between 0 and 3 except the one which is the age of patient. There are six different disease classification in the original data but our technique only support binary classification therefore the data was altered so that the classification is either Psoriasis (the most common of disease classification) or not Psoriasis.

None of the above datasets have aggregate outputs. They are traditional datasets in that they contain an output value for each input vector. Therefore, we used them for experiments by creating aggregate training sets. We grouped together multiple input vectors in the training set and aggregate their output values together. This technique for creating artificial aggregate datasets gives us the capability to run multiple experiments, each with different characteristics. Specifically, we vary the dataset in two different ways, each of which could potentially influence the performance of an aggregate output learning algorithm. First, we vary the size of the aggregate sets, i.e., the number of rows in the original dataset whose outputs are summed to form each aggregate set. For the traditional supervised learning problem, all aggregate sets are of size 1. Note that the set size is essentially an upper bound on how much information is lost due to aggregation. For simplicity, all aggregate sets that we generate for a particular dataset have the same size. Second, we vary the amount of randomness in the aggregation. We do this to check whether the results depend on the type of collections given. The ”randomness“ value seen in our experimental results refers to the number of pairs that we randomly swapped before aggregating the data.

We used the quadratic programming solver CPLEX to handle the optimization. We varied the parameter C and D in order to achieve the right balance of weights of ξ, z in the formulation given in eq. (3). We found in our experiments that the value of C is the ratio of the examples of class B to the total no. Of examples and the value of D is the ratio of the examples of class A to the total no. of examples. The parameter E, given in formulation of step (2) does not affect the accuracy much. The algorithm may have to perform different no. of iteration of step(2) for different values of E but converges to the same solution.

We measured the accuracy of our algorithm on the same dataset, where in the training set, output values are aggregated over multiple input vectors while in the test set, each input vector has its own output value.Table2 shows the results of our algorithm for the “*Breast-cancer-wisconsin*” dataset. Table 3 shows the results for the “*Dermatology*” dataset.

	Randomness							
Size	**0**	**25**	**50**	**100**	**200**	**500**	**1000**	**2000**
2	95%	95.57%	95.57%	94.42%	94.28%	95.71%	94.14%	95.28%
5	94%	93.85%	94%	94%	94.14%	95.57%	94.28%	95.28%
10	92.85%	93.57%	93.28%	93.28%	93.57%	93.57%	94.14%	92.14%
20	93%	93.85%	94.42%	94%	94.14%	96.0%	94.14%	93.28%

Table2: Accuracy on dataset “*Breast-Cancer-wisconsin*”

	Randomness							
Size	**0**	**25**	**50**	**100**	**200**	**500**	**1000**	**2000**
2	98.61%	98.33%	97.5%	100%	98.61%	98.33%	99.44%	98.61%
5	98.33%	98.33%	98.33%	97.22%	97.22%	98.88%	96.38%	99.44%
10	98.33%	99.44%	98.05%	97.77%	96.66%	97.5%	94.44%	95.83%
12	94.44%	94.16%	92.5%	93.61%	96.94%	98.33%	89.44%	90.27%
15	89.44%	90.27%	93.33%	96.94%	95.27%	99.44%	89.44%	95.00%

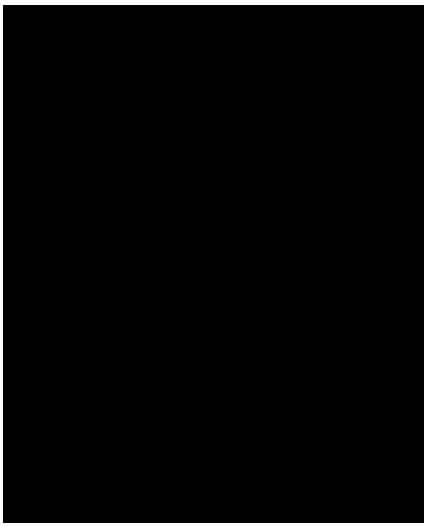
Table3: Accuracy on dataset "*Dermatology*"

Now, we compare our SLAAG algorithm with the MIQP formulation given by Musicant et al in [1] with respect to time and accuracy in both datasets. Table 4 shows the accuracy comparison between these two approaches for the varying size of collections. As the table 4 shows that the accuracy given by these two approaches are comparable.

As discussed earlier that the main drawback of Mixed Integer quadratic program is the amount of time it takes to give solution, so here we compare the time taken by the MIQP formulation and SLAAG algorithm. Fig 1 and Fig 2 show the time taken by these two approaches in both the datasets for different no. of examples. Figures clearly show that the time taken by MIQP formulation increases drastically as the number of examples increase and become impractical for large datasets. Whereas our algorithm gives solution magnitudes of times faster while also maintaining sufficient accuracy.

Breast-cancer-wisconsin			**Dermatology**		
Size	**MIQP**	**SLAAG**	**Size**	**MIQP**	**SLAAG**
2	**95%**	**95.28%**	**2**	**98%**	**98.61%**
5	**93%**	**95.28%**	**5**	**99%**	**99.44%**
10	**93%**	**92.14%**	**10**	**98%**	**95.83%**

Table4: Accuracy comparison between MIQP Formulation and SLAAG Algorithm



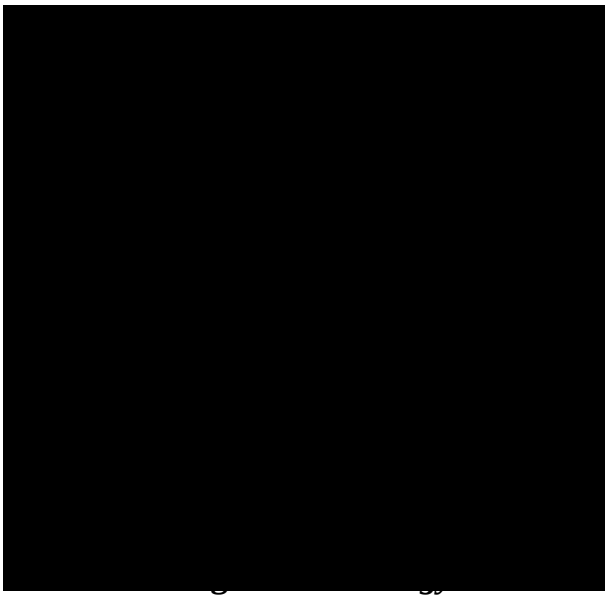


Fig 1: *breast-cancer-wisconsin*

6. Conclusion:

We have proposed a concave formulation for the aggregate output learning problem [1] and given a much faster finite approximation algorithm based on linear programming. Unlike the MIQP formulation, our algorithm can handle large datasets. Our experiments show that in spite of being much faster than MIQP formulation, our linear algorithm also maintains sufficient accuracy. Hence our algorithm is an efficient and viable tool for handling large datasets. In future, our work can be extended for multi class and non linear version of the problem.

7. References

[1] D. R. Musicant, J. M. Christensen, and J. F. Olson, "Supervised learning by training on aggregates outputs," *in proc. Of the Seventh Int'l Conference on Data Mining.* IEEE Press ,2007, pp.252-261

[2] K. Bennett and A. Demiriz. "Semi-supervised support vector machines" *in Advances in Neural Information Processing Systems,* volume 12, pages 368–374. MIT Press, 1998.

[3] G. Fung and O. Mangasarian. "Semi-supervised support vector machines for unlabeled data classification" *Optimization Methods and Software*, 15:29–44, 2001.

[4] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases, 1992 ,*http://www.ics.uci.edu/∼mlearn/MLRepository.html*.

[5] O.L. Mangasarian. "Machine learning via polyhedral concave minimization". *Applied Mathematics and parallel computing*, pages 175 – 188, 1996.

[6] O.L. Mangasarian and P.S.Bradley "Feature selection via concave minimization and support vector machine" *Machine learning proceedings of the Fifteenth International Conference* , pages 82-90 , 1998