Learning SVM from Distributed, Non-Linearly Separable Datasets with Kernel Methods

Karlen Mkrtchyan

***Abstract*— Learning from distributed data sets is common problem nowadays and the question of its actuality can be inferred by the number of applications and from even higher number of problems coming from real world business solutions. Here we will review the question of distributed classification with Support Vector Machines, and present our approach to handle the problem in effective way.**

***Index Terms*— Machine Learning, SVM, Distributed SVM, Kernel Methods, Distributed Computation.**

I. Introduction

Learning from distributed data is actual problem nowadays, and there are many methods for coping with this problem with various machine learning algorithms including SVM, Decision Trees, LPC, Neural Networks and etc. Here we specifically address to SVM, and the justification of that comes from its practical use. Whether in which cases, what classification method should be used is beyond our topic, but practice shows that there is no such algorithm that covers all needs.

Formally, classification problem is the following. We are given a dataset S, a hypothesis set H and in some cases performance criterion P. The chosen learning algorithm L as an output gives $h \in H$. Dataset $S$is the set of labeled training examples, and each of them is n dimensional vector, where each component is drawn from predefined value space. Here, for clearance, we will review only real valued examples. The aim for learning h is that we want to predict label for future unlabeled examples.

In distributed environment the problem statement changes. In these cases we have $S_1, S_2, \ldots S_k$ datasets, and set of restrictions R, which may be empty in some cases. R is meant to represent unique constraints of environment, for example communication cost, data privacy and etc. The aim is to construct SVM, so that it will be the same as if we constructed SVM on $\cup_{i=1}^{k} S_k$, or it will approximate SVM($\cup_{i=1}^{k} S_k$) with any ε accuracy.

II. Support Vector Machines

Assume we are given

$$S = \{(x_i, y_i) | i = 1,2,\ldots,\} \subset R^d \times Y$$

Where x_i belongs to R^d and label $y_i \in Y$. The aim is to construct hypothesis $h : R^d \to y$ so that $h(x_i)$ will be close to y_i for predicting$h(x)$ for unknown S. For simplicity, we will assume that $Y = \{+1, -1\}$ which is called binary classification problem.

Karlen Mkrtchyan, Karlen Mkrtchyan, PhD student, Institute for Informatics and Automation Problems (IIAP), Armenia

Main principle of SVM is to separate the given space by hyper plane, but in practice given dataset is not linearly separable, because of this, we consider the case when dataset is not linearly separable. This problem in centralized systems solved with kernel methods. That is, the given$d$ dimensional space mapped into higher dimensional space $R^{'}$ with the function$\phi$where sample $x$becomes$\phi(x)$. Practically we are interested in production of $\phi(x_i)\phi(x_j)$ and not in exact value of $\phi(x)$. The method as result outputs new linearly separable dataset. The problem of SVM in linear separable case is as follows

$$\min_{\alpha \in R^l} = \sum_{i=1}^{l} a_i + \sum_{i=1}^{l} y_i\, y_j\, a_i a_j\, x_i x_j \qquad (1)$$

$$s.t \sum_{i=1}^{l} a_i\, y_i = 0$$

$$0 < a_i < C\,, i = 1,2 \ldots l$$

It is known constraint optimization problem. It is properly proved by applying KKT [1] conditions in SVM base problem [1]. Let $K(x^T, z) R^d * R^d \to R$ be an inner product kernel function which satisfies Mercer's condition. We will construct the nonlinear map function in the way that $K(x_i^T, x_j) = \phi(x_i^T)\phi(x_j)$, which means we don't have to give the function ϕexplicitly, because only inner product is needed. SVM with kernel function becomes the following problem.

$$h(x) = \sum_{u_j \neq 0} u_j K(x_i^T, x_j) + b$$

III. Distributed Approach

Here we will assume, that each of the datasets $S_i, i = 1,..k$, is linearly separable. Firstly let's review the case when the $\cup_{i=1}^{k} S_k$ is linearly separable as well.

The basic approach, that is learn SVM in distributed system separately, and then combine support vectors. This approach is easy to interpret and cost effective, only support vector transfer is needed. Although practically it works, but (Caragea, Silvescu & Honavar 2000) showed that $SV(\cup S_k) \neq SV(\cup SV(S_k))$, consequently it's not exact solution to even the simple case when $\cup_{i=1}^{k} S_k$ is linearly separable.

In [4] is shown that the convex hulls of the instances that belong to the two classes is sufficient for learning SVMs from distributed data. Let $Conv(S)$ be the convex hull of set S. The algorithm calculates convex hulls $Conv(S_i(+))$ and $Conv(S_i(-))$ for all $i = 1,2,\ldots k$ and sends it to computation center. Then on the union of positive and negative convex hull centralized SVM is applied. It's also shown in (Gruber & Wills 1993) that

$$Conv(\bigcup_{i=1}^{k} S_k) = \bigcup_{i=1}^{k} Conv(S_k)$$

Which solves the problem of exact SVM, but it is not effective, as long as convex hulls may be too big and its size growing exponentially depending on data dimensionality. It can be seen that we can minimize this transfer by dividing the exchanging procedure. To do this at each step we will calculate support vectors step by step that is we calculate support vectors at one point, then, broadcast these vectors to all participants. Every participant connects this support vector to its data set and builds SVM, then it broadcasts its vectors to all participants, this process continues until at some point all participants have the same support vectors. It is obvious that this process converges, because in worst case participants will exchange all their data, which is practically not feasible in case when $\bigcup_{i=1}^{k} S_k$ is linearly separable. In fact what we do here is constructing support vectors that would be convex hull of $\bigcup_{i=1}^{k} S_k$, step by step construction allows to be sure that set of support vectors will be minimum, because at each step we add support vectors to our global set only if it is required for some of participants and only in that case.

In practice there are many cases when $\bigcup_{i=1}^{k} S_k$ is not linearly separable. And in this case by merely combining support vectors we cannot achieve any reasonable results. Here we purpose the following approach. Firstly, we construct SVMs separately at each point. To combine these SVMs we can use kernel trick. To do this we map set of hyper plans to a new space, in a way that the mapping reflects shape of hyper planes with hyper plane in new space. Below figures show the visual example of this case

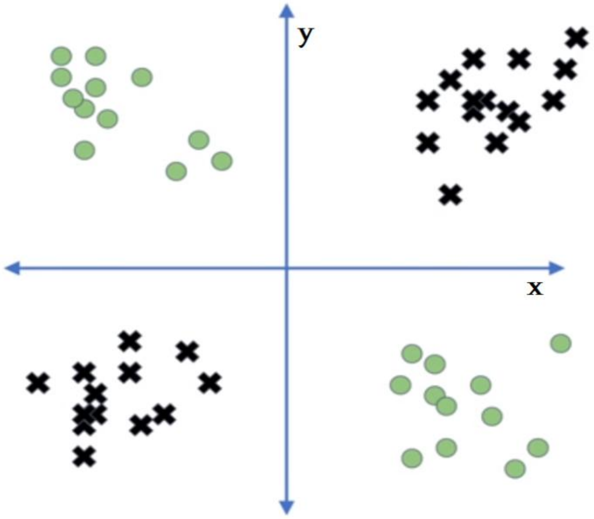

Figure 1. Linearly *non separable sets, which can be divided into two linearly separable sets*

To handle the problem, we must find kernel function that would map these points to new space where data is linearly separable. It is known that choosing SVM kernel function is something that achieved by employing kernels, and the best one is chosen by experiments, that is there is no universal best Kernel function. Below is the visualization of the results we want to achieve

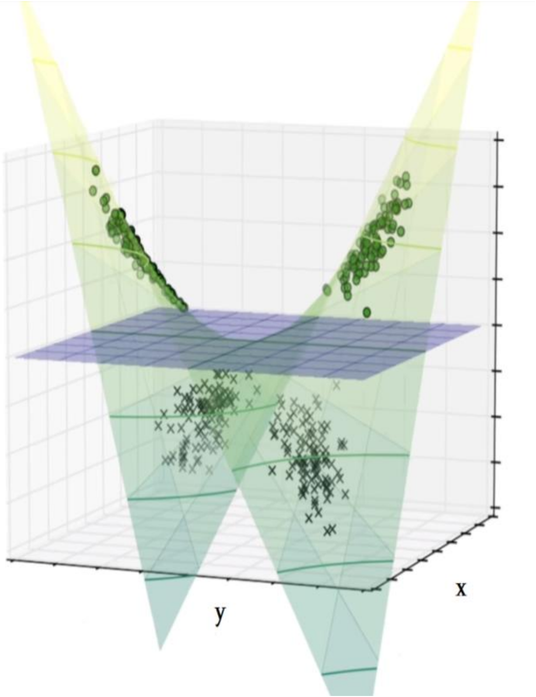

Figure 2. *mapping data points to higher dimensional space with kernel tricks*

It is obvious that the kernel function can be constructed by hand, and the final result would be the same as if we had data in one place or at least we can approximate its accuracy by any ε. Practically, the solution with distributed data gives huge advantage in efficiency. Kernel functions calculate huge matrixes which is dependent on data samples count, by calculating kernel function only on support vectors reduces computation cost, because number of support vectors are much smaller that dataset power. In [6] there is result considered family of Gaussian Kernels which proved to be simple in interpretation and very flexible. To interpret kernel function, the following should be done. All we need to do is to map the points of one class in positive axis of new space, and to the opposite to new second one. It is obvious that it can be done for each dataset, because they are already linearly separable. It is also conceivable that mere combining of these mappings will give us all dataset mapped in new space with exact separation of classes. And thus we will have not one, but the vector of kernel functions, at each one corresponding to one of given datasets.

IV. Conclusion

Results provided here that the nature of SVM gives wide specter of improvisations in distributed environments. It is possible to achieve exactly the same results as in centralized cases, and even with fewer computations. Further investigation could be done to make it clear the form of kernel functions used in the case, when we join linearly separable case, despite Gaussian Kernels are comparably useful here, but these kernels are not designed to this specific case. These are our first steps towards the effective solution in distributed environments with this approach.

REFERENCES

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[9] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[10] (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume(issue). Available: http://www.(URL)

[11] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

[12] (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)

[13] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876—880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar