Distributed SVM

Luojie Xiang

I. Introduction

The idea of the project is to design and implement a distributed SVM with the following two requirements:

- Faster training than standard SVM
- Better resiliency towards malicious attack

The attack model and attack algorithms are described in next two sections.

II. Attack Model

This work assumes all computers can be trusted. The attack comes from the training dataset. The adversary has the following capabilities:

- Know the victim's dataset.
- Has computing power to create data points.
- Can insert data points into the victim's dataset.

This is a practical assumption. The difference of attackers comes in the algorithm they use to create data points. Different algorithms may result in different damage power on the victim's SVM accuracy.

Next session looks into attack algorithms.

III. Attack Algorithm

The attacks include some or all of the following:

- Loss Optimization attack - create datapoints that optimize loss function [10]
- Red herring attack - add fake feature and have target classifier depend on fake features as heavily as possible [11]
- Inseparability attack - mix the features from target classifier into same data points to confuse it [11]
- Furthest-First Flip attack - pick the furthest data point from the decision plane and flip its label [8]

A. Red Herring Attack

This attack creates fake features and a strong statistical relationship between the fake features and the class label. This encourages the SVM to learn the fake features as important or indicative features, so that the decision will rely heavily on them. The distribution of the fake features will of course be very different from other benign features. Thus, the prediction accuracy of SVM will be compromised.

The attack works as follows:

- Step 1 - Create N data points, each with k random benign features.
- Step 2 - Cut the N data points into 2 halves, S_1 and S_2. For all points in S_1, add a feature that never appeared in the benign dataset.
- Step 3 - Label all points in S_1 as positive and those in S_2 as negative.
- Step 4 - Insert the data points (interleaved, one positive one negative) into victim's dataset.

B. Inseparability Attack

This attack mixes the features of the victim's training set and give them random labels. This will confuse the victim's SVM and bring down its accuracy.

The attack works as follows:

- Step 1 - Create N data points, each with k random benign features.
- Step 2 - For each data point, assign a random label.
- Step 3 - Insert the data points into victim's dataset.

C. Furthest-First Flip Attack

This attack picks the data point that is furthest from the SVM decision plane and flip its label. It is shown to generate a near-optimal attack effect on SVM [8]. Another benefit of this attack over the optimal attack is, it works on discrete feature space, since optimal attack involves solving convex optimization over loss function which requires a differentiable field. This limits optimal attacks to only continuous feature space.

The attack works as follows:

- Step 1 - Train an SVM on benign data set.
- Step 2 - Apply the decision function of SVM on all data points of the benign data set.
- Step 3 - Sort the decision values and find the maximum in absolute value.
- Step 4 - Flip the chosen point and insert it back into the benign data set.

IV. SVM Training Algorithm

The design of SVM training algorithm should be parallelizable so that the training algorithm could be faster and has better tolerance against malicious attacks than standard SVM algorithm.

1. Baseline: Ensemble Learning

Ensemble Learning is a classic idea of improving any machine learning algorithm's resilliency against noise in the training dataset [1], [3]. It has also been used as a defense strategy against malicious attacks [2], [4].

Ensemble learning is used as our baseline, in which the training happens as follows:

- Step 1 - Create N samples out of the training set by sampling with replacement.
- Step 2 - Scatter N samples to N machines.
- Step 3 - Train an SVM on each machine.

- Step 4 - All N SVMs vote on new data points during prediction.

2. A better Algorithm: Modified Adaptive Cluster

Adaptive Cluster was proposed for reducing training set size for SVM [5], [6], based on the idea that, SVM's decision plane relies only on support vectors [7]. The process is,

- Step 1 - Cluster the training set.
- Step 2 - Train an initial SVM based on the representatives of the clusters (such as centroid).
- Step 3 - Train a final SVM using the clusters that contain support vectors in the initial SVM.

This method is modified to include resiliency against malicious attacks.

- Change 1 - After step 1 of original Adaptive Cluster algorithm, add a sanitization step, using either distance filtering, or active learning (explained later).
- Change 2 - Use the Adaptive Cluster algorithm in the ensemble framework. Step 3 in ensemble learning algorithm is changed to Adaptive Cluster algorithm above.

Distance filtering - Malicious data points will be far away from benign data points. For example, label flip attack [8] picks benign data, say positive class, flip the label to negative class and insert it back to the training set. These data points will reside in the positive area except now they all have negative label. Therefore, the malicious points (with negative label) will be very far away with the benign negative points. Thus, they're very likely to be put into a separate cluster if the negative points are clustered and this cluster is very likely to be very far away from the other negative clusters. Using some distance filtering, such as probability distribution threshold will identify the malicious cluster.

Active Learning - Active learning have machine learning algorithms summarizes the dataset and present to human experts the data points that it is most not sure of [9]. Human experts will label these very concise summaries and guide the learning process. Since in Adaptive Cluster algorithm, the representatives of each cluster is found, they can be convieniently used as the summary. Human expert will tell the learning algorithm which representatives are malicious. The cluster whose representative is labeled malicious by the expert will be thrown away.

V. Evaluation

The evaluation comes in three fold:

Speed - The speed would be compared with standard SVM training.

Resiliency against malicious attacks - The accuracy will be compared with standard SVM as more malicious points are added.

A. Dataset

A lot of datasets are made publicly available thanks to competitions such as KDD CUP, Kaggle and research groups such as UCI Machine Learning Repository, LIBSVM [12], [13], [14], [15]. Appropriate dataset will be selected later to facilitate testing.

References

[1] Breiman, Leo. "Bagging predictors." Machine learning 24, no. 2 (1996): 123-140.

[2] Barreno, Marco, Peter L. Bartlett, Fuching Jack Chi, Anthony D. Joseph, Blaine Nelson, Benjamin IP Rubinstein, Udam Saini, and J. Doug Tygar. "Open problems in the security of learning." In Proceedings of the 1st ACM workshop on Workshop on AISec, pp. 19-26. ACM, 2008.

[3] Dong, Yan-Shi, and Ke-Song Han. "Boosting SVM classifiers by ensemble." In Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 1072-1073. ACM, 2005.

[4] Cretu, Gabriela F., Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. "Casting out demons: Sanitizing training data for anomaly sensors." In Security and Privacy, 2008. SP 2008. IEEE Symposium on, pp. 81-95. IEEE, 2008.

[5] Boley, Daniel, and Dongwei Cao. "Training Support Vector Machines Using Adaptive Clustering." In SDM. 2004.

[6] Yu, Hwanjo, Jiong Yang, and Jiawei Han. "Classifying large data sets using SVMs with hierarchical clusters." In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 306-315. ACM, 2003.

[7] Koggalage, Ravindra, and Saman Halgamuge. "Reducing the number of training samples for fast support vector machine classification." Neural Information Processing-Letters and Reviews 2, no. 3 (2004): 57-65.

[8] Xiao, Han, Huang Xiao, and Claudia Eckert. "Adversarial Label Flips Attack on Support Vector Machines." In ECAI, pp. 870-875. 2012.

[9] Raghavan, Hema, Omid Madani, and Rosie Jones. "Active learning with feedback on features and instances." The Journal of Machine Learning Research 7 (2006): 1655-1686.

[10] Battista Biggio, Blaine Nelson, Pavel Laskov: Poisoning Attacks against Support Vector Machines. ICML 2012

[11] Newsome, James, Brad Karp, and Dawn Song. "Paragraph: Thwarting signature learning by training maliciously." In Recent advances in intrusion detection, pp. 81-105. Springer Berlin Heidelberg, 2006.

[12] http://www.sigkdd.org/kddcup/index.php

[13] http://www.kaggle.com/

[14] http://archive.ics.uci.edu/ml/

[15] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

[16] John C. Platt. "Fast training of support vector machines using sequential minimal optimization". In Bernhard Schlkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, 1998. MIT Press.

Dataset	Source	# of classes	# of data (training/testing)	# of features (training/testing)
w8a	[16]	2	49,749 / 14,951	300 / 300