Data Poisoning for Linear Models

Adhyyan Narang, Rohan Sinha, Anand Siththaranjan, Forest Yang*
Department of Electrical Engineering and Computer Science
University of California, Berkeley

Abstract

Despite its empirical success, the field of machine learning, and in particular deep learning, is plagued by the existence of train and test time adversarial attacks. The former problem, known as data poisoning, considers an adversary who has the ability to alter the training data-set prior to learning to coerce the learner to learn a corrupted model. While empirical attacks have been created for neural networks using iterative methods, we are not aware of any data poisoning attacks that are accompanied by certificates of optimality. Towards this end, we attempt to construct certifiably optimal data poisoning attacks for linear models using principled optimization techniques to formulate efficiently solvable convex problems. For some uncertainty sets, these exactly solve the problem; in others, we obtain experimentally verifiable approximations of the optimal poisoning attack. We test on synthetic data to show the efficacy of our techniques and discuss how the intuitive takeaways from this work can apply to data-poisoning attacks in general.

1 Introduction

Deep learning models have been shown to perform remarkably well at a wide variety of tasks, from using imitation learning to allow self-driving cars to drive in traffic [Bojarski et al., 2016] to surpassing humans and mastering the game of Go [Silver et al., 2016]. However, recent experiments [Shafahi et al., 2018] have revealed that the safety of these systems can be easily compromised by ill-intentioned adversaries in practice, through attacks which add or modify training points. This type of attack, called *data poisoning*, was first investigated by Biggio et al. [2012]. The inability to ensure robust performance in practice makes deploying these models in safety-critical settings quite risky, and discourages the widespread adoption of these cutting-edge models.

In 2014, Szegedy and Fergus [2014] demonstrated that it is simple to construct adversarial test-set attacks for most neural networks. Learned models that otherwise achieve superhuman performance are comprehensively and reliably fooled by imperceptible perturbations to test images. This surprising discovery has inspired research into procedures to defend against these test-set attacks.

However, training set attacks [Biggio, 2011] are comparatively understudied even thought they have the potential to be just as catastrophic for machine learning models. In these training time attacks, referred to as “data poisoning," an adversary can manipulate some subset of the training data prior to the learning process to influence the test-time predictions of the learned model. The details of the abilities and intention of the adversary depend on the specific application and problem formulation.

In practice, large datasets are often collected from external sources for data-driven applications. Therefore, an adversary could easily insert a small number of carefully picked examples into training data to alter the learned model. For example, recent work showed that it was possible to introduce correctly labeled data of frogs into training data to trick the learned model into misclassifying pictures of other animals at testing time [Shafahi et al., 2018]. However, the attacks mentioned were

*All authors contributed equally to this work

Preprint. Under review.

constructed for neural networks and had no theoretical guarantees. The aim of this paper is to study the effect of data-poisoning attacks on more transparent learning setups, primarily ridge regression, where it is possible to take a more principled and theoretical approach.

Contributions and paper outline

Our contributions are as follows:

- Formulate the problem of data poisoning in the general case and remark its relationship to the robust optimization (adversarial training) problem.
- For convex uncertainty sets, propose a relaxed convex problem, and experimentally verify that the quality of the relaxation is good.
- For a spectral-norm constrained perturbation matrix, we exactly reduce the problem to one solvable using line-search over a scalar variable.

In addition to being interesting and practically relevant in its own right, we expect that data poisoning attacks against linear models will yield intuitive insight about strategies for attackers that can motivate defenses for more complex setups, such as deep learning, as well.

Our paper is organized as follows. First we provide context through related works in Section 2. Then, we clarify preliminaries and notation in Section 3, and define our problems of interest in Section 4. We proceed to consider solutions to the data poisoning problem for linear models in Section 5, and empirically verify the quality of our relaxations in Section 6. In Section 7, we discuss the intuitive interpretations of our results and proposed directions of future exploration. Implementations of our attacks and experiments can be found on our project website[2].

2 Related Works

2.1 Poisoning Attacks on Neural Networks

Shafahi et al. [2018] investigated how to construct data poisoning attacks to impair test performance of neural networks, in situations where the attacker does not have the ability to change the labels of the dataset (often referred to as a clean-label attacks). To create an attack which perturbed input images, the authors took advantage of feature collisions resulting from representations of the input in the later layers of neural networks by creating other adversarial inputs that have the same representation. This is accomplished by using a two-part iterative optimization procedure to first find an adversarial point and then project it back into the space of realistic input. Although we consider clean-label attacks as well, we approach constructing an attack for simpler linear models where principled optimization techniques can be more readily applied. By formulating the data poisoning problem for specific linear models, we can more effectively take advantage of the structure of these learning algorithms to formulate approximate methods to solve the underlying optimization problem.

2.2 Stability of Feature Selection Methods under Poisoning Attacks

Xiao et al. [2015] consider the effect of data poisoning on the classification accuracy and stability of popular feature selection methods such as LASSO, ridge regression and elastic net. The proposed method involved alternating between gradient updates with restrictions based on the Karush Kuhn Tucker conditions to increase loss on certain points, and learning a model on this new training data. In our work, we allow for perturbations on all points and formulate convex problems that approximate the optimal perturbation without needing to relearn a new model. We also suggest a method that can recover the optimal poisoned dataset when considering largest singular value perturbations.

2.3 Certified defenses for data poisoning attacks

Steinhardt et al. [2017] develop certifiable defenses to data poisoning attacks under assumptions on train/test set concentration and outliers in the dataset. While this work considers defenses and experiments with neural networks, the present work concerns only attacks and in linear models.

[2]Project website: https://github.com/AnandS29/DataPoisoning

3 Notation

Consider input space $\mathcal{X}$, output space $\mathcal{Y}$, function class $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$, and loss $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$. For an arbitrary predictor $f \in \mathcal{F}$, the risk is its expected loss with respect to the underlying distribution of the data: $R(f) = \mathbb{E}_{x,y}\ell(f(x), y)$. Since we cannot compute the true risk since we do not know the underlying data distribution, we define the empirical risk $\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$, an unbiased estimator of the true risk. A classifier is learned via empirical risk minimization: $\hat{f} = \text{argmin}_{f\in\mathcal{F}} \hat{R}(f)$.

We will primarily consider the setting of Euclidean inputs $\mathcal{X} = \mathbb{R}^d$ and linear predictors $\mathcal{F} = \mathbb{R}^d$. For linear regression, $\mathcal{Y} = \mathbb{R}$ and for logistic regression, $\mathcal{Y} = \{0, 1\}$. Let $X_0 = [x_1 \ \ldots, \ x_n]^\top \in \mathbb{R}^{n\times d}$ denote our train matrix and y our vector of labels. $x_i \in \mathbb{R}^d$ is our ith training point and $y_i \in \mathcal{Y}$ is its corresponding label.

We assume the adversary can modify the input data by choosing it to be an arbitrary member of a perturbed data matrix set $\mathbf{X}$, i.e. $X \in \mathbf{X}$. Equivalently, we assume there is an unaltered version of the data, $X_0 \in \mathbb{R}^{n\times d}$, and that the poisoner can add a perturbation $\Delta = [\delta_1 \ \ldots \ \delta_n]^\top \in \mathbb{R}^{n\times d}$ in some perturbation set $\mathbf{\Delta}$, so $X = X_0 + \Delta$ and $\mathbf{X} = X_0 + \mathbf{\Delta}$. Typically, $\mathbf{\Delta}$ will be defined by a bound on a certain matrix norm, such as: $\mathbf{\Delta} = \{\Delta \in \mathbb{R}^{n\times d} : \max_{1\le i\le n} \sqrt{\sum_{j=1}^{d} \Delta_{ij}^2} \le \epsilon\}$ or operator norm: $\mathbf{\Delta} = \{\Delta \in \mathbb{R}^{n\times d} : \|\Delta\|_2 \le \epsilon\}$.

4 The General Problem of Data Poisoning

Making the training set hard to fit

In this work, we consider a data poisoning scenario where the adversary wants to prevent the learner from learning anything at all. To do this, the goal of the adversary is to maximize the training loss:

$$p^* = \max_{X\in\mathbf{X}} \min_{\theta\in\Theta} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) \tag{1}$$

If, we exchange the max and min above to obtain the dual:

$$d^* = \min_{\theta\in\Theta} \max_{X\in\mathbf{X}} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) \tag{2}$$

which is exactly the traditional robust training problem. This result explicates a relationship between the problems of *constructing a data poisoning attack* and *training a model that is robust to a test-set attack*, which have previously been studied completely independently.

In this work, we focus on constructing a data poisoning attack for ridge regression. We also note a possible method for generating these adversarial datasets for the case of L_2 regularized logistic regression, which is noted in the appendix (section 10.5).

5 Poisoning Ridge Regression

5.1 Concrete problem

The standard ridge regression problem is as follows:

$$p^*_{\text{nominal}} = \min_w \|Xw - b\|_2^2 + \lambda\|w\|_2^2 \tag{3}$$

By using standard adversarial formulations of this problem, we can construct the robust problem as:

$$p^*_{\text{robust}} = \min_w \max_{X\in\mathbf{X}} \|Xw - b\|_2^2 + \lambda\|w\|_2^2 \tag{4}$$

The dual of this robust problem can be viewed as formulation of the data poisoning problem for ridge regression:

$$d^*_{\text{robust}} = p^*_{\text{poison}} = \max_{X \in \mathbf{X}} \min_{w} \|Xw - b\|_2^2 + \lambda \|w\|_2^2 \tag{5}$$

5.2 Relaxation to convex problem

The inner minimization of problem 5 can be solved in closed form to provide the following non-convex problem in X:

$$= - \min_{X \in \mathbf{X}, t, U} t - b^T b \ : \ \begin{bmatrix} U & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \ , \ U = X^T X + \lambda I \tag{6}$$

We propose using an SDP relaxation of the above problem such that an upper bound on the primal solution can be achieved:

$$p^*_{\text{poison}} \leq \tilde{p}^*_{\text{poison}} = - \min_{X \in \mathbf{X}, t, U} t - b^T b \ : \ \begin{bmatrix} U - \lambda I & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \ , \ \begin{bmatrix} U & X^T \\ X & I \end{bmatrix} \succeq 0 \tag{7}$$

Refer to the appendix (section 9.1) for a full proof.

5.3 Improving the relaxation

From the problem 7, we note that the objective can be reduced by simply making U arbitrarily large. This makes the quality of our relaxation worse. To combat this, we suggest two methods to constrain this variable such that our relaxed solution is more accurate:

- Adding a regularizing term to objective
- Constructing bounds on U based on those for X

Trace regularization

Here, we simply add regularizing term $\mu \text{Tr}(U)$ to the objective to prevent the values of U from becoming too large.

$$\tilde{p}^*_{poison} = - \min_{X \in \mathbf{X}, t, U} t - b^T b + \mu \text{Tr}(U) \ : \\ \begin{bmatrix} U & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \ , \ \begin{bmatrix} U - \lambda I & X^T \\ X & I \end{bmatrix} \succeq 0$$

To find the optimal μ, we employ traditional techniques like cross-validation to find it.

For independent uncertainty on training points: Interval Bounds on U

We can improve the above SDP relaxation (with or without the slack cost on the trace of U) by bounding U based on the feasible set of perturbations on X. To do this, we assume we can perturb the each datapoint within some convex disturbance set. In this report, we primarily consider perturbations with bounded infinity norm:

$$x_i \in \{x \in \mathbb{R}^n | x = \hat{x}_i + \delta_i, \|\delta_i\|_\infty \leq \rho\} \ \ \forall i \in [1, m] \tag{8}$$

In the infinity norm case, this equivalent to an alternate definition of the perturbation set on the columns of X:

$$\mathcal{X} = \left\{ X_0 + [\delta_1 \quad \dots \quad \delta_n]^\top \ | \ \forall i = 1, \dots, n, \ \|\delta_i\|_\infty \leq \rho \right\}. \tag{9}$$

Note that before relaxing the problem, U is equal to:

$$U = X^T X + \lambda I \tag{10}$$

$$= \hat{X}^T \hat{X} + \Delta^T \hat{X} + \hat{X}^T \Delta + \Delta^T \Delta + \lambda I \tag{11}$$

This clearly shows that each entry of U lies between some lower and upper bound $U_{i,j} \in [\underline{U}_{i,j}, \bar{U}_{i,j}]$ where:

$$\underline{U}_{i,j} = \min_{\|\delta_i\|_\infty, \|\delta_j\|_\infty \leq \rho} \hat{c}_i^T \hat{c}_j + \delta_i^T \hat{c}_j + \hat{c}_i^T \delta_j + \delta_i^T \delta_j + \lambda \mathbb{1}\{i = j\} \tag{12}$$

$$\bar{U}_{i,j} = \max_{\|\delta_i\|_\infty, \|\delta_j\|_\infty \leq \rho} \hat{c}_i^T \hat{c}_j + \delta_i^T \hat{c}_j + \hat{c}_i^T \delta_j + \delta_i^T \delta_j + \lambda \mathbb{1}\{i = j\} \tag{13}$$

As is shown in the appendix (section 9.2) these problems have solutions that can be rapidly computed. This gives us a simple method to improve the relaxation of the data poisoning problem for ridge regression under infinity norm perturbations. However, for perturbation sets bounded by different norms the row and column formulations are not equivalent. In this case, we can use the kernelized formulation of ridge regression combined with additional approximations to find useful bounds on U. This is because XX^T, a series of inner products between datapoints, appears in kernel ridge regression. A procedure to bound U in kernelized regression is outlined in the appendix (section 9.3).

5.4 Special case: Uncertainty on spectral norm of training matrix

In the case where we consider a largest singular value constraint, we are able to solve the poisoning problem exactly, and also propose a line search method for efficient computation (detailed in appendix 9.4). The highlights are as follows.

We define the uncertainty on X defined by the following set:

$$\mathbf{X} = \{X \in \mathbb{R}^{n \times d} : \|X - X_0\|_2 \leq \epsilon\}$$

With this choice of uncertainty set, we can construct an equivalent non-convex QCQP that is independent of X:

$$\max_{\mu} -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T \mu \ : \ \|X_0^T \mu\|_2 \leq \epsilon \|\mu\|_2$$

As there is a single quadratic constraint, the S-procedure can be applied here to formulate a convex SDP. The original poisoned X can be recovered by performing a line search algorithm derived in the appendix. The algorithm is to first let $U \in \mathbb{R}^{n \times n}$ be the left singular vectors of X_0 and $\sigma_1, \ldots, \sigma_n$ the corresponding singular values; set $d = U^\top y$, and line search for an r^* satisfying

$$\sum_{i=1}^{n} \frac{\sigma_i^2 d_i^2}{(1 + r^* \sigma_i^2)^2} = \epsilon^2 \sum_{i=1}^{n} \frac{d_i^2}{(1 + r^* \sigma_i^2)^2}.$$

Then, we set

$$\lambda^* = \sqrt{\sum_{i=1}^{n} \frac{d_i^2}{(1 + r^* \sigma_i^2)^2}}$$

and set $c^* = \lambda^* d + \lambda^* r^* \Sigma^2 d$ where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n \times n}$. Then we can recover $\mu^* = Uc^*$, and $X^* = PX_0$, where $P \in \mathbb{R}^{n \times n}$ is the orthogonal projection onto $\mathrm{span}\{\mu^*\}^\perp$.

6 Experiments

6.1 Setup

We ran experiments that tested the following variants of our data poisoning attacks on synthetic data:

- Convex relaxed problem without bounded U
- Regularized problem
- Bounded U problem

To construct the synthetic data, we made a random nominal $X_0 \in R^{n \times d}$ and $w^* \in \mathbb{R}^d$ from which we can make $y = X_0 w^* + \mathcal{N}(\mu, \sigma)$. We tested on a dataset with 100 data points and 10 features ($n = 100$, $d = 10$) and used noise parameterized by $\mu = 0$ and $\sigma = 0.1$.

We conduct two types of comparison to show the effectiveness of our attacks:

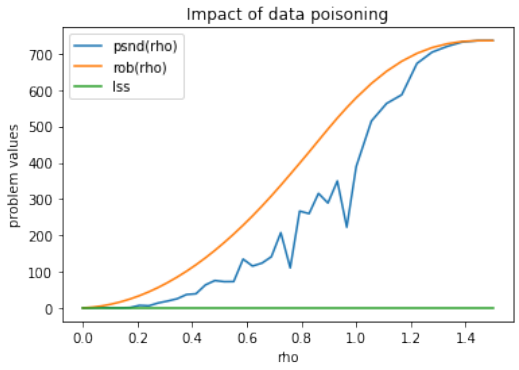

Figure 1: psnd(rho) is value of optimal least squares solution on poisoned data (obtained from initial relaxation), rob(rho) is an upper bound on psnd(rho), lss is least squares value on original data.

6.1.1 Training loss

We compare the training loss, or the objective value, for the nominal least-squares problem, the robust problem and the various relaxed problems under different levels of perturbation to the nominal data matrix X_0. Specifically, the values are derived from the following:

$$\begin{aligned}
\text{lss} &= \min_{w \in \mathbb{R}^d} \|X_0 w - y\|_2^2 \\
\text{rob}(\rho) &= \min_{w \in \mathbb{R}^d} \max_{X \in \mathbf{X}(\rho)} \|Xw - y\|_2^2 \\
\text{psnd}(\rho) &= \min_{w \in \mathbb{R}^d} \|X_{psnd}(\rho) w - y\|_2^2
\end{aligned}$$

We note that the robust problem is an upper bound on the relaxed problems, giving us a notion of the quality of our approximation.

6.1.2 Loss on nominal X_0

We also look at the loss on the actual X_0, which is formed from finding the loss of a dataset under the least-squares objective using a w learnt from the poisoned dataset. This is computed using the following:

$$\begin{aligned}
w_{\text{psnd}}(\rho) &= \operatorname*{argmin}_{w \in \mathbb{R}^d} \|X_{\text{psnd}}(\rho) w - y\|_2^2 \\
\text{lss}_{\text{psnd}} &= \|X_0 w_{\text{psnd}}(\rho) - y\|_2^2
\end{aligned}$$

6.2 Initial effectiveness of relaxation

We use our initial convex relaxed method on the constructed dataset to undertake the following experiment.

We note for comparison that the average norm of rows is 3.06. This signifies that applying a relatively small perturbation, e.g. norm 0.5, to each row drastically increases the smallest loss the learner can achieve on the data, from less than 10 to almost 100. Furthermore, while $\text{psnd}(\rho) \leq \text{rob}(\rho)$ (as should be the case by weak duality), we note they follow a comparable trend. That is, by poisoning the dataset and letting the learner do their best, the attacker can do comparable damage to if they could wait for the learner to choose a predictor and then choose the worst perturbation.

In addition to examining each problem's objective value i.e. the training loss, we examine the loss achieved by the resulting linear predictors on the original dataset. This can be considered a form of test set evaluation for the poisoning attack – we are seeing how badly the predictor learned from a poisoned dataset does on the original dataset. This is shown in the below plot, for each value of the perturbation size ρ.

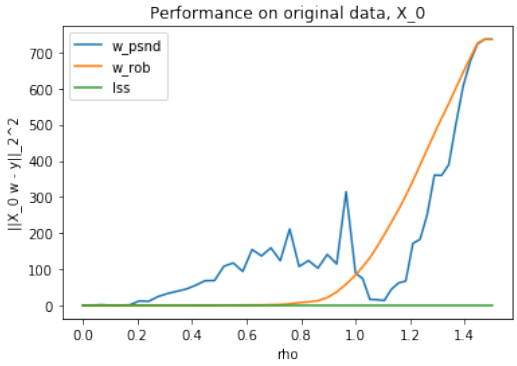

Figure 2: w_psnd learned via least squares on poisoned data (from initial relaxation); w_rob via robust problem

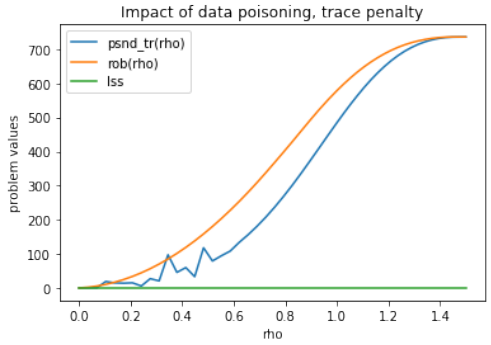

Figure 3: Same as figure 1, except poisoned data obtained from relaxation with trace regularization

We see that the robust solution w_{rob} is a good solution to the original linear regression problem, since for $\rho \leq 0.8$ it performs close to optimally on the unperturbed data. This matches our intuition, because w_{rob} is intended to perform well on the original data, and in addition be robust.

On the other hand, the poisoned solution w_{psnd} does badly on the original linear regression problem, even for small values of ρ. This indicates that data poisoning which only maximizes loss on the poisoned data can disrupt test performance, i.e. significantly increase loss on the original data. Interestingly, when ρ reaches about 1.0, the poisoned solution does well on the original data. This warrants further investigation. Another observation from this plot is that in convex $\min \max$ problems, the solutions obtained from switching the $\min$ and $\max$ are usually similar, but here, this is not the case. In other words, robust w's and poisoned w's may be different.

6.3 Using trace regularization

Recall this method adds the penalty term $\mu \operatorname{Tr} U$ to the objective, which in a sense makes the relaxation of the poisoning problem closer to the original, since it deals with the unboundedness of U. The same experiments as previously but with the penalized psnd_{tr} solutions with μ set to 0.01 are seen in Figures 3 and 4.

We note that the data poisoning curves are smoother overall, and closer to the robust curve in the first plot. This suggests that the poisoner benefits from trace regularization, and also that the true data poisoning problem values form a smooth curve which closely follows the robust problem values.

The performance of the predictor learned from poisoned data on the original data corroborates the same general trend described in the previous section.

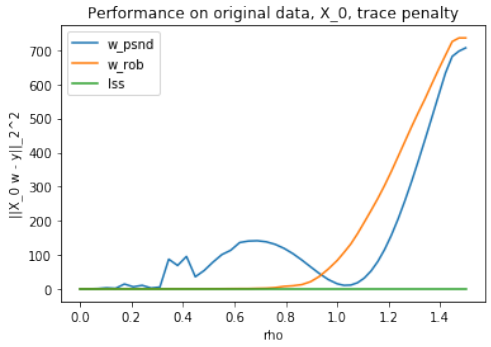

Figure 4: Same as figure 2, except poisoned data obtained from relaxation with trace regularization

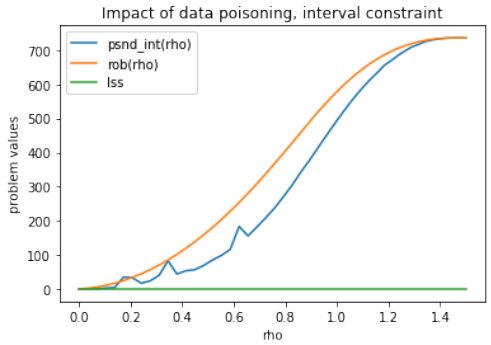

Figure 5: Same as figure 1, except poisoned data obtained from relaxation with interval bounds

6.4 Using bounded U relaxation

This approximation places an interval constraint on U based on the perturbation set $\mathbf{X}$; this refines the relaxation by safely shrinking the feasible set. Here are the same experiments with the refined psnd_{int} solutions:

Similar to the previous method, we see that in certain regions the resulting curve is smoother than the original method which just relaxes the problem without placing constraints on U.

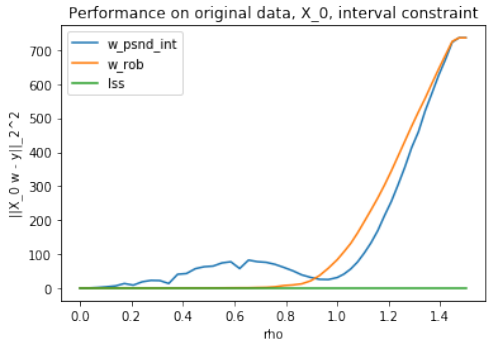

Figure 6: Same as Figure 1, except poisoned data obtained from relaxation with interval bounds

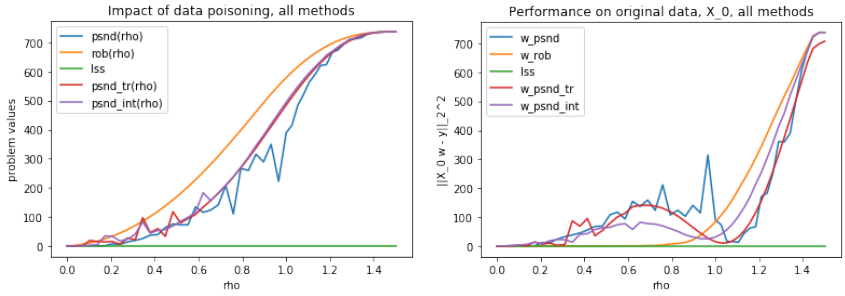

Figure 7: All methods plotted on same plot

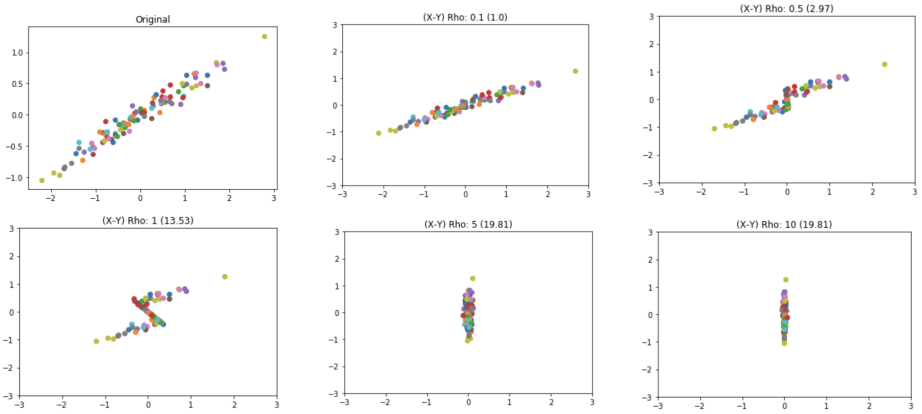

Figure 8: Effect of Data-Poisoning on 1-D linear regression. The training data is plotted on the x-axis, the training labels on the y-axis. As ρ increases, the adversary perturbs the samples to decrease the rank of the data.

7 Discussion

7.1 Low rank training matrices

As the experiments showed, small perturbations to the training data can cause large training losses. This shows that the data rapidly becomes more difficult to fit by linear functions. In figure Figure 8, we show how the data is modified by the adversary for a linear regression problem with one dimensional data ($x, y \in \mathbb{R}$). As we increase ρ, the size of the perturbation set available to the adversary increases, and we slowly see the data resembles the original sample less and less. What is particularly interesting, is that the optimal perturbation seems to project the data into a lower-rank space. For the 1-D case, we can see that all the gets moved towards the its mean, 0. It makes sense that reducing the rank of the data is the optimal strategy for the adversary, since for linear models the predictions must lie within the span of the data-features.

7.2 Limitations

Though the proposed problems are convex semi-definite programs and are as such solvable in polynomial time, the semidefinite programs can still take a while to solve. In particular, solving an instance of the SDP relaxation with $n = 100, d = 10$ takes around 20 seconds. Furthermore, the relaxation suffers from the unboundedness of U, which can be ameliorated to an extend with trace regularization and interval constraints.

7.3 Future Directions

7.3.1 Impairing test-time performance

Although our experiments showed that a model learned on poisoned data can significantly degrade in performance when faced with test data from the original distribution, we did not explicitly force this behavior. In data poisoning attacks that impair testing time performance, the train performance is not necessarily impaired. But the unaware learner has chosen a classifier that has certain deficiencies at test time. In the following, let $f_{\hat{\theta}(X)}$ denote the classifier learned by running an algorithm on input data X. For example, for linear regression,

$$f_{\hat{\theta}(X)} = \operatorname*{argmin}_{w \in \mathbb{R}^d} \|Xw - y\|_2.$$

Impair performance on specific test point

Intuitively, what we want to measure is how badly the performance of the learned model on a test point (x_t, y_t) is affected by data poisoning, i.e.:

$$\max_{X \in \mathbf{X}} \ell(f_{\hat{\theta}(X)}(x_t), y_t). \tag{14}$$

This is similar to Shafahi et al. [2018], which constructed adversarial inputs to the learning algorithm to cause misclassification. Formalizing the idea of targeted test set attacks for our approach would be practically useful to study adversarial applications of this method.

Impair performance in expectation

Another interesting quantity is simply the performance of the algorithm with perturbed input on a test point from the same distribution as the original input. That is, if $\forall i = 1, \ldots, n,\ (x_i, y_i) \sim \mathcal{D}$, then we examine

$$\max_{X' \in X + \mathbf{\Delta}} \mathbb{E}_{(x,y)\sim\mathcal{D}}(f_{\hat{\theta}(X')}(x), y). \tag{15}$$

We can approximate the expectation by the empirical expectation. Thus our problem becomes, partitioning the data into $X = X_1 \cup X_2$ and letting $x_1, \ldots, x_n$ denote the data points in X_2:

$$\max_{X \in X_1 + \mathbf{\Delta}} \sum_{i=1}^{n} \ell(f_{\hat{\theta}(X)}(x_i), y_i).$$

7.3.2 Different Attacker Abilities

In this work, we considered an attacker that was able to perturb any data point with a constraint on the amount of perturbation. This is plausible when the attacker has access to the initial training dataset. However, in other settings there may be limitations on what the attacker is able to do. For example, they may only be able to perturb k data points, or possible can only add adversarial data points to an existing data set. As such, future works may want to consider how to adjust the above optimization procedures to account for these differences.

7.3.3 Stability of Feature Selection Methods

As seen in Xiao et al. [2015], we note the relationship between data poisoning and the selection of stable features (where under perturbation to the training set the features chosen by feature selection methods do not vary too much). As such, further work can analyse how our proposed attacks impact stability by applying metrics such as Kuncheva's stability index [Kuncheva, 2007] or others detailed in Nogueira et al. [2018]. This would provide another notion apart from training and test error for how well our proposed attacks perform in affecting learning algorithms. It may also motivate work in directly trying to optimize against an algorithm to encourage feature instability.

7.3.4 Transferability to other learning algorithms

Though the poisoning attacks detailed in this paper are constructed to be adversarial to their respective learning algorithms, it is possible that they may still affect different learners. It is possible to see that for other learning algorithms such as support vector machines (SVM), we can possible make problems infeasible in the case of hard-margin SVMs or increase training error for soft-margin SVMs. In the case of universal function approximators like decision trees and neural networks, though we would be unable in reducing training loss we hypothesize that the complexity of these models would increase. As well, if we were able to bound the complexity of these learner, such as by placing limits on the depth of the decision tree or the norm of the weight matrices for a neural network, we can then see if there is any degradation in performance.

8 Conclusion

We posed the general problem of data poisoning aimed at disrupting train time performance, focusing on the cases of linear and logistic regression. This is a max-min problem that can be viewed as a game where the order the learner and attacker pick strategies is switched, compared to the typical min-max robustness problem. However, upon switching to the max-min data poisoning problem, the problem becomes non-convex and difficult to solve. We showed that we can approximate the optimal poisoned data matrix for linear and logistic regression using optimization techniques such as Schur complements and semidefinite programming. Furthermore, we show empirically that we can improve the approximation with trace regularization and interval constraints derived from the perturbation set. If the perturbation set is defined by bounded spectral norm, we show that the S-procedure can be used to compute the exact problem value with semidefinite programming. Finally, we give a way to recover the optimal poisoned dataset in this case with a line search. Our experiments also suggest that poisoning the data, which in our definition has as an explicit goal of worsening the training time performance, also disrupts test time performance. Future work includes adapting our methods to actively worsening performance on a particular test dataset from learning on the poisoned train dataset.

Acknowledgements

We would like to thank Professor Laurent El Ghaoui, Professor Somayeh Sojoudi and Armin Askari for their continued support and guidance during the process of completion of this project.

References

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1467–1474, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1. URL http://dl.acm.org/citation.cfm?id=3042573.3042761.

Nelson B Laskov P Biggio, B. Support vector machines under adversarial label noise. *IEEE Transactions on Knowledge and Data Engineering*, 20, 2011. URL http://pralab.diee.unica.it/en/node/751.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv e-prints*, art. arXiv:1604.07316, Apr 2016.

Ludmila I. Kuncheva. A stability index for feature selection. In *Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, AIAP'07, pages 390–395, Anaheim, CA, USA, 2007. ACTA Press. URL http://dl.acm.org/citation.cfm?id=1295303.1295370.

Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the stability of feature selection algorithms. *Journal of Machine Learning Research*, 18(174):1–54, 2018. URL http://jmlr.org/papers/v18/17-514.html.

Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. *arXiv e-prints*, art. arXiv:1804.00792, Apr 2018.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016. ISSN 0028-0836. doi: 10.1038/nature16961.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. *CoRR*, abs/1706.03691, 2017. URL http://arxiv.org/abs/1706.03691.

Zaremba W. Sutskever I. Bruna J. Erhan D. Goodfellow I. Szegedy, C. and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6199.

Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1689–1698, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/xiao15.html.

Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Mach. Learn.*, 85(1-2):41–75, October 2011. ISSN 0885-6125. doi: 10.1007/s10994-010-5221-8. URL https://doi.org/10.1007/s10994-010-5221-8.

9 Appendix

9.1 Solving and relaxing poisoned ridge regression problem

The optimal solution of ridge regression is known to be the following:

$$p^*_{\text{nominal}} = -b^T X(X^T X + \lambda I)^{-1} X^T b + b^T b$$

The inner minimization of the above problem can be solved in closed form to provide the following non-convex problem in X:

$$\begin{aligned}
p^*_{\text{poison}} &= \max_{X \in \mathcal{X}} -b^T X(X^T X + \lambda I)^{-1} X^T b + b^T b \\
&= -\min_{X \in \mathcal{X}} b^T X(X^T X + \lambda I)^{-1} X^T b - b^T b \\
&= -\min_{X \in \mathcal{X}, t} t - b^T b \ : \ t \geq b^T X(X^T X + \lambda I)^{-1} X^T b \\
&= -\min_{X \in \mathcal{X}, t} t - b^T b \ : \ \begin{bmatrix} X^T X & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \\
&= -\min_{X \in \mathcal{X}, t, U} t - b^T b \ : \ \begin{bmatrix} U & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \, , \, U = X^T X
\end{aligned}$$

This problem contains a quadratic matrix equality, hence it is not convex. We propose relaxing this problem into a convex problem by changing the equality constraint $U = X^T X$ into $U - X^T X \succeq 0$ and construction a linear matrix inequality by using Schur complements:

$$\begin{aligned}
p^*_{\text{poison}} \leq \tilde{p}^*_{\text{poison}} &= -\min_{X \in \mathcal{X}, t, U} t - b^T b \ : \ \begin{bmatrix} U & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \, , \, U - X^T X \succeq 0 \\
&= -\min_{X \in \mathcal{X}, t, U} t - b^T b \ : \ \begin{bmatrix} U & X^T b \\ b^T X & t \end{bmatrix} \succeq 0 \, , \, \begin{bmatrix} U & X^T \\ X & I \end{bmatrix} \succeq 0
\end{aligned}$$

9.2 Constructing interval bounds on U

Consider the following optimization problems:

$$\underline{U}_{i,j} = \min_{||\delta_i||_\infty, ||\delta_j||_\infty \leq \rho} \hat{c}_i^T \hat{c}_j + \delta_i^T \hat{c}_j + \hat{c}_i^T \delta_j + \delta_i^T \delta_j + \lambda \mathbb{1}\{i = j\} \tag{16}$$

$$\bar{U}_{i,j} = \max_{||\delta_i||_\infty, ||\delta_j||_\infty \leq \rho} \hat{c}_i^T \hat{c}_j + \delta_i^T \hat{c}_j + \hat{c}_i^T \delta_j + \delta_i^T \delta_j + \lambda \mathbb{1}\{i = j\} \tag{17}$$

Note that since these problems have infinity norm constraints, this decouples each component from each other. For simplicity, we will consider the minimization problem directly. We can rewrite this problem as follows:

$$\underline{U}_{i,j} = \sum_{k=1}^{n} \min_{|\delta_{i,k}|, |\delta_{j,k}| \leq \rho} \hat{c}_{i,k}^T \hat{c}_{j,k} + \delta_{i,k}^T \hat{c}_{j,k} + \hat{c}_{i,k}^T \delta_{j,k} + \delta_{i,k}^T \delta_{j,k} + \lambda \mathbb{1}\{i = j\} \tag{18}$$

Note that these inner 1-D optimization problems are of the form:

$$p^* = \max ax + by + xy \tag{19}$$

$$|x|, |y| \leq \rho \tag{20}$$

These problems are not convex, but have the following property:

Theorem 1. *For any value of (a,b), the optimal value $(x*, y*)$ of this optimization is always on one of the corners of the feasible box i.e $|x^*| = |y^*| = \rho$.*

Proof. Assume $y^* = \alpha sign(b)$. Then the objective becomes

$$ax + \alpha|b| + x\alpha sign(b) = \alpha|b| + x(a + \alpha sign(b)).$$

We can clearly see that to maximize this, if $(a + \alpha sign(b)) < 0$, then $x = -\rho$ and otherwise $x = \rho$. So, for any fixed value y^*, x^* will be one of the corners of the feasible set. This argument is symmetrical in the decision variables of the optimization, so it holds for y^* as well. □

Therefore, we can rapidly compute the solutions to the inner minimization problems in equation 18 by searching over the corners of the 2-D perturbation box. This result symmetrically extends to the maximization problem as well. Therefore, we can efficiently construct upper and lower bounds on U in $\mathcal{O}(d^2 n)$ time.

9.3 Kernelized Ridge Regression solution

The inner minimization of the above problem can be solved in closed form to provide the following non-convex problem in X:

$$\underset{w}{\operatorname{argmin}} ||Xw - y||_2^2 + \lambda ||w||_2^2 = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda \mathbb{I})^{-1} y$$

Plugging this back into our poisoning problem, we obtain:

$$\begin{aligned}
p^*_{\text{poison}} &= \max_{X \in \mathcal{X}} -y^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{X}^T y + y^T y && (21)\\
&= -\min_{X \in \mathcal{X}} +y^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{X}^T y - y^T y && (22)\\
&= -\min_{X \in \mathcal{X}, t} t - b^T b \ : \ t \geq y^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{X}^T y && (23)\\
&\leq -\min_{X \in \mathcal{X}, t} t - b^T b \ : \ \begin{bmatrix} \mathbf{X}\mathbf{X}^T + \lambda\mathbb{I} & XX^T y \\ y^T & t \end{bmatrix} \succeq 0 && (24)\\
&= -\min_{X \in \mathcal{X}, t, \mathbf{K}, \widetilde{\mathbf{K}}} t - b^T b \ : \ \begin{bmatrix} \widetilde{\mathbf{K}} & \mathbf{K}y \\ y^T & t \end{bmatrix} \succeq 0 \ , \ \widetilde{\mathbf{K}} = \mathbf{X}\mathbf{X}^T + \lambda\mathbb{I}, \mathbf{K} = \mathbf{X}\mathbf{X}^T && (25)
\end{aligned}$$

In the above, we remark that 24 is a safe approximation to 23, justifying the inequality above. Consider the following justification for this.

Lemma 1.

$$\frac{1}{2}\begin{bmatrix} \boldsymbol{X}\boldsymbol{X}^T + \lambda\mathbb{I} & \boldsymbol{X}\boldsymbol{X}^T y \\ y^T & t \end{bmatrix} \succeq 0 \implies t \geq y^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\mathbb{I})^{-1}\boldsymbol{X}\boldsymbol{X}^T y$$

Proof. First, realize that $\forall A, x^T A x = x^T A^T x$. Hence, $\forall x, x^T A x \succeq 0 \implies \forall x, x^T(A + A^T)x \succeq 0$. Hence, our assumption above implies that

$$\begin{bmatrix} \mathbf{X}\mathbf{X}^T + \lambda\mathbb{I} & \frac{y^T + \mathbf{X}\mathbf{X}^T y}{2} \\ \frac{y^T + \mathbf{X}\mathbf{X}^T y}{2} & t \end{bmatrix} \succeq 0$$

By Schur complements, we have that

$$\frac{1}{4}(y^T + \mathbf{X}\mathbf{X}^T y)\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I}^{-1}(y^T + \mathbf{X}\mathbf{X}^T y) \leq t.$$

By expanding and applying the inequality $a^T a + b^T b \geq a^T b$, we remark that

$$\frac{1}{4}(y^T + \mathbf{X}\mathbf{X}^T y)\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I}^{-1}(y^T + \mathbf{X}\mathbf{X}^T y) \geq y^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{X}^T y$$

Our result directly follows. □

The problem 25 contains a quadratic matrix equality, hence it is not convex. We relax this problem into a convex problem similar to previously by changing the equality constraints into conic inequality constraints instead.

$$p^*_{\text{poison}} \leq \tilde{p}^*_{\text{poison}} = -\min_{X \in \mathcal{X}, t, \mathbf{K}, \widetilde{\mathbf{K}}} t - b^T b \ : \ \begin{bmatrix} \widetilde{\mathbf{K}} & \mathbf{K}y \\ y^T & t \end{bmatrix} \succeq 0 \ , \ \widetilde{\mathbf{K}} \succeq \mathbf{X}\mathbf{X}^T + \lambda\mathbb{I}, \mathbf{K} \succeq \mathbf{X}\mathbf{X}^T$$

We can now use Schur complements to represent these as LMI's.

$$p^*_{\text{poison}} \leq \tilde{p}^*_{\text{poison}} = -\min_{X \in \mathcal{X}, t, \mathbf{K}, \widetilde{\mathbf{K}}} t - b^T b \ : \ \begin{bmatrix} \widetilde{\mathbf{K}} & \mathbf{K}y \\ y^T & t \end{bmatrix} \succeq 0 \, , \begin{bmatrix} \widetilde{\mathbf{K}} - \lambda\mathbb{I} & \mathbf{X}^T \\ \mathbf{X} & \mathbb{I} \end{bmatrix} \succeq 0, \begin{bmatrix} \mathbf{K} & \mathbf{X}^T \\ \mathbf{X} & \mathbb{I} \end{bmatrix} \succeq 0$$

This relaxed optimization problem is a convex semi-definite program.

9.4 Solution to Spectral Norm Constrained Data Poisoning for Least-Squares Regression

First, we add a constraint $Xw = z$, which will allows us to later take the dual:

$$\min_w \|Xw - y\|_2^2 = \min_{w,z} \|z - y\|_2^2 \ : \ z = Xw$$

We note without proof that strong duality holds by Slater's condition:

$$\max_\mu \min_{w,z} \|z - y\|_2^2 + \mu^T(z - Xw)$$
$$= \max_\mu \min_w -w^T(X^T\mu) + \min_z z^T z + \mu^T z - 2y^T z$$

For the sub-problem over z, we find the optimal value of z^* to be $y - \frac{1}{2}\mu$. For the other minimization problem over w, we note the optimal value is 0 if $X^T\mu = 0$ and $-\infty$ otherwise. Thus the outer maximization would optimize for μ to be in the nullspace of X^T. This results in the following problem:

$$\max_\mu -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T\mu \ : \ X^T\mu = 0$$

If we consider the original poisoning problem, $\max_{X\in\mathcal{X}} \min_w \|Xw - b\|_2^2$, for a largest singular value constraint this results in the following maximization problem:

$$\max_{X,\mu} -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T\mu \ : \ X^T\mu = 0 \ , \ \|X - X_0\|_2 \le \epsilon$$

9.4.1 Reformulation to QCQP with single constraint

Though the norm constraint on X is convex, the constraint $X^T\mu = 0$ is non-convex in X and μ. However, we can show that the constraints together imply a certain convex constraint.

We can rewrite the above problem as:

$$\max_\mu -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T\mu \ : \mu \in M$$

where $M = \{\mu : \exists X : X^T\mu = 0, \|X - X_0\|_2 \le \epsilon\}$ since the objective is independent of X. Now consider the following reformulation of the feasible set.

Lemma 2. $M' := \{\mu : \|X_0^T\mu\|_2 \le \epsilon\|\mu\|_2\} = \{\mu : \exists X \ s.t \ X^T\mu = 0, \|X - X_0\|_2 \le \epsilon\} := M$

Proof. **Backward direction.**

For $\mu \in M$, we show that $\mu \in M'$.

Using $\|X - X_0\|_2 \le \epsilon$ and $X^\top\mu = 0$:

$$\|X_0^T\mu\|_2 = \|(X - X_0)^T\mu\|_2 \le \|X - X_0\|_2\|\mu\| \le \epsilon\|\mu\|.$$

Forward direction. Now, for $\mu \in M'$, we show that $\mu \in M$.

To simplify our calculation, We note that

$$\|(X - X_0)u\|_2^2 \le \epsilon^2\|u\|_2^2 \ , \ \forall u \ \equiv \ \|(X - X_0)^T v\|_2^2 \le \epsilon^2\|v\|_2^2 \ , \ \forall v$$

Define $P \in \mathbb{R}^{n\times n}$ to be the orthogonal projection map onto the subspace orthogonal to μ. Let $X^T = X_0^T P$, and let any $t \in \mathbb{R}^n$ be decomposed as $t = au + b\mu$, where $u \in \{c\mu : c \in \mathbb{R}\}^\perp$, i.e. $au = Pt$. Then,

$$\frac{\|(X^T - X_0^T)t\|_2}{\|t\|_2} = \frac{\|X_0^T Pt - X_0^T t\|_2}{\|t\|_2} = \frac{\|aX_0^T u - (aX_0^T u + bX_0^T\mu)\|_2}{\|t\|_2}$$
$$= \frac{|b|\|X_0^T\mu\|_2}{\|t\|_2} \le \frac{|b|\|X_0^T\mu\|_2}{\|b\mu\|_2} = \frac{\|X_0^T\mu\|_2}{\|\mu\|_2} \le \epsilon$$

This shows that X satisfies the norm constraint as $\frac{\|(X^T - X_0^T)t\|_2}{\|t\|_2} \leq \epsilon, \forall t$. As well, it is clear by the construction of P that $X^T\mu = X_0^T P\mu = 0$. □

9.4.2 Applying S-lemma to reformulate into SDP

We know have the following problem after changing constraints:

$$\max_{\mu} -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T\mu \ : \ \|X_0^T\mu\|_2 \leq \epsilon\|\mu\|_2$$

As this is a non-convex QCQP with only a single quadratic constraint, we can apply the S-lemma here and can consider the following equivalent convex optimization problem:

$$= \min_{t} t \ : \ -y^T y + \frac{1}{2}\mu^T y - \frac{1}{4}\mu^T\mu \leq t \ , \ \forall \mu \text{ s.t } \|X_0^T\mu\|_2 \leq \epsilon\|\mu\|_2$$

$$= \min_{t,\lambda} t \ : \ \lambda \begin{bmatrix} X_0X_0^T - \epsilon^2 I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \succeq \begin{bmatrix} -\frac{1}{4}I & \frac{1}{4}y \\ \frac{1}{4}y^T & t - \|y\|^2 \end{bmatrix}$$

9.4.3 Recovering optimal μ^* and X^*

We refer to Appendix B in Boyd and Vanderberghe, which details an equivalence between the original non-convex QCQP and the following convex minimization problem (if the non-convex problem is strictly feasible):

$$(U^*, u^*) = \arg\min_{U,u} \frac{1}{4}\mathbf{tr}(U) - \frac{1}{2}u^T y + y^T y \ : \ \mathbf{tr}((X_0X_0^T - \epsilon^2 I)U) \leq 0 \ , \ \begin{bmatrix} U & u \\ u^T & I \end{bmatrix} \succeq 0$$

It is shown in Boyd and Vanderberghe that the original rank one constrained problem that we are trying to solve has value equal to the above SDP relaxation. Therefore, an optimal solution is $U^* = \mu^*\mu^{*\top}$, $u^* = \mu^*$ where μ^* solves the original problem. Hence we can solve for the optimal u^* and thus recover the optimal μ^*.
Given μ^*, an optimal X^* is recovered by taking $X^* = X_0P$, where P is the projection map on the subspace orthogonal to μ^*. By finding this P, which is tractable, we can construct an optimal poisoned dataset X^*.

9.5 LSV Algorithm: Line search method

We give a line search algorithm to solve the problem

$$\min_{u\in\mathbb{R}^n} \frac{1}{4}u^\top u - \frac{1}{2}u^\top y + y^\top y \ : \ \|X_0^\top \frac{u}{\|u\|}\| \le \epsilon. \tag{26}$$

We note that it is equivalent to solve the problem

$$\min_{u\in\mathbb{R}^n} -u^\top y \ : \ \|u\| = 1, \ \|X_0^\top u\| \le \epsilon. \tag{27}$$

This is because the constraint in (26) is invariant to scaling, and so, letting $\hat{u}$ with $\|\hat{u}\| = 1$ be a feasible solution to (26), any $u = a\hat{u}$ for $a \in \mathbb{R}$ is feasible. The optimal value is

$$\min_{a\in\mathbb{R}} \frac{a^2}{4} - \frac{\hat{u}^\top y}{2}a + y^\top y = -\frac{(\hat{u}^\top y)^2}{4} + y^\top y.$$

This proves that we can replace the objective with $-\frac{(u^\top y)}{4} + y^\top y$ and restrict $\|u\| = 1$, which leads to (27), since $(\cdot)^2$ is monotonic in the absolute value of its input.
Now using the SVD, we finally note our problem equals

$$\min_{c\in\mathbb{R}^n} -c^\top d \ : \ \|c\| = 1, \ \|\Sigma c\| \le \epsilon, \tag{28}$$

where $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix containing the singular values of X_0 and $c = U^\top u$, $d = U^\top y$, where $U \in \mathbb{R}^{n\times n}$ are the left singular vectors of X_0. If we solve (28) for c^*, then $u^* = Uc^*$.

9.5.1 Convert to line search

There are two cases for (28). The first case is trivial. We know that $\min_{\|c\|=1} -c^\top d = -\|d\|$, so if $\|\Sigma\frac{d}{\|d\|}\| \le \epsilon$ we are done: just take $c^* = \frac{d}{\|d\|}$.
In the second case, we can deduce that the optimal c must satsify both $\|c\| = 1$ and $\|\Sigma c\| = \epsilon$, i.e. the inequality is an equality. This follows from properties of optimization with equality and inequality constraints:

Proof. If the inequality is not tight at optimality: $\|\Sigma c^*\| < \epsilon$, then by complementary slackness, the optimal dual variable $\nu_2^* = 0$. Then, by stationarity of the Lagrangian ($-c^\top d + \nu_1(c^\top c - 1) + \nu_2(c^\top\Sigma^2 c - \epsilon^2)$), we obtain

$$d = 2\nu_1^* c^*.$$

This implies we are in the first case: $c^* = \frac{d}{\|d\|}$. But this is a contradiction. Thus, $\|\Sigma c\| = \epsilon$. □

The above observation lets us convert the inequality constraint to an equality, which means we can forgo a 2-d grid search in favor of a line search.
This is how the line search works. By Lagrangian stationarity,

$$-d + \lambda c + \mu\Sigma^2 c = -\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} + \begin{bmatrix} (\lambda+\mu\sigma_1^2)c_1 \\ (\lambda+\mu\sigma_2^2)c_2 \\ \vdots \\ (\lambda+\mu\sigma_n^2)c_n \end{bmatrix} = 0 \implies \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} d_1/(\lambda+\mu\sigma_1^2) \\ d_2/(\lambda+\mu\sigma_2^2) \\ \vdots \\ d_n/(\lambda+\mu\sigma_n^2) \end{bmatrix}.$$

This means our constraints are actually

$$\sum_{i=1}^n \frac{d_i^2}{(\lambda+\mu\sigma_i^2)^2} = 1$$
$$\sum_{i=1}^n \frac{\sigma_i^2 d_i^2}{(\lambda+\mu\sigma_i^2)^2} = \epsilon^2.$$

If we define $r = \frac{\mu}{\lambda}$, our constraints are

$$\sum_{i=1}^{n} \frac{d_i^2}{(\lambda + \lambda r \sigma_i^2)^2} = 1$$
$$\sum_{i=1}^{n} \frac{\sigma_i^2 d_i^2}{(\lambda + \lambda r \sigma_i^2)^2} = \epsilon^2.$$

Instead searching over a 2-d grid for λ and μ, we can instead just search over r, because r determines λ.

$$\sum_{i=1}^{n} \frac{d_i^2}{(\lambda + \lambda r \sigma_i^2)^2} = 1 \implies \sum_{i=1}^{n} \frac{d_i^2}{(1 + r \sigma_i^2)^2} = \lambda^2 \implies \sum_{i=1}^{n} \frac{\sigma_i^2 d_i^2}{(1 + r \sigma_i^2)^2} = \epsilon^2 \sum_{i=1}^{n} \frac{d_i^2}{(1 + r \sigma_i^2)^2}.$$

So our algorithm is just to line search for an $r \in \mathbb{R}$ which satisfies

$$\sum_{i=1}^{n} \frac{\sigma_i^2 d_i^2}{(1 + r \sigma_i^2)^2} = \epsilon^2 \sum_{i=1}^{n} \frac{d_i^2}{(1 + r \sigma_i^2)^2}.$$

From that r^*, we can recover λ^*, then $\mu^* = \lambda^* r^*$, then $c^* = (\lambda^* + \mu^* \Sigma^2) d$, then $u^* = U c^*$.

9.6 Poisoning Logistic Regression

We regard the nominal L_2 regularised logistic regression problem as the following:

$$p^*_{\text{nominal}} = \min_w -\sum_{i=1}^{n} y_i \ln s(w^T X_i) + (1-y_i)\ln(1-s(w^T X_i)) + \lambda \|w\|_2^2$$

where $s(\gamma) = \frac{1}{1+e^{-\gamma}}$, the logsitic function.

Similar to the ridge regression case, we can formulate the robust and poisoned counterpart of this problem:

$$p^*_{\text{robust}} = \min_w \max_{X \in \mathbf{X}} -\sum_{i=1}^{n} y_i \ln s(w^T X_i) + (1-y_i)\ln(1-s(w^T X_i)) + \lambda \|w\|_2^2$$

$$p^*_{\text{poisoned}} = \max_{X \in \mathbf{X}} \min_w -\sum_{i=1}^{n} y_i \ln s(w^T X_i) + (1-y_i)\ln(1-s(w^T X_i)) + \lambda \|w\|_2^2$$

By recognizing the equivalence between maximum entropy and regularized logistic regression [Yu et al., 2011], we can formulate the following optimization problem:

$$\max_{\alpha, X \in \mathbf{X}} -\frac{1}{2}\alpha^T Z Z^T \alpha - \alpha^T \log(\alpha) - (1-\alpha)^T \log(1-\alpha)$$

where $Z = \text{diag}(y)X$. We omit a derivation of the proof of this result in this paper.

This problem is not jointly convex in α and X, however it is bi-convex as it is convex in each variable separately. As such, we can try to find a solution to this problem by iteratively solving the two convex problems till convergence.