Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners
Shike Mei and Xiaojin Zhu
{mei, jerryzhu}@cs.wisc.edu

Take-Home Message
- We play "white hat hackers".
- We optimally poison the training set to mislead machine learners to specific wrong models.
- This is done via a bilevel optimization framework and KKT conditions.

Identifying Attacks by the KKT Conditions

For convex and regular objective $O_L$ and continuous search space $\mathcal{D}$ (e.g., continuous features space), we reduce the framework to a single-level constrained optimization problem via the Karush–Kuhn–Tucker (KKT) conditions of the lower-level problem $\min_{D,\theta} O_L(D,\theta)$ s.t.

$$\begin{align*}
\lambda^T_i (\theta) &= 0, \quad i = 1, \ldots, m, \\
\mu^T_i (\theta) &= 0, \quad i = 1, \ldots, m.
\end{align*}$$

We optimize training data $D$ by projected gradient descent. In the $t$-th iteration, we update the data from $D(t)$ to $D(t+1)$ by

$$D(t+1) = \text{proj}_{\mathcal{D}} \left( D(t) + \alpha_t \nabla O_L(D(t),\theta(t)) \right),$$

where $\nabla O_L(D(t),\theta(t)) = \nabla^2 O_L(D(t),\theta(t)) \frac{\partial^2}{\partial D \partial \theta}$. We assume that $\nabla^2 O_L(D(t),\theta(t))$ can be easily calculated. To calculate $\frac{\partial^2}{\partial D \partial \theta}$, we denote $i = (\theta, \lambda, \mu)$ and calculate $\frac{\partial^2}{\partial D \partial \theta}$ by the implicit function theorem

$$\frac{\partial}{\partial D} \frac{\partial}{\partial \theta} \left| \frac{\partial}{\partial \theta} \right| = \frac{\partial}{\partial \theta} \left| \frac{\partial}{\partial \theta} \right| \frac{\partial}{\partial D} \left| \frac{\partial}{\partial \theta} \right|.$$

Where $f = 0$ represents the equality constraints in KKT conditions and

$$f(D,\theta,\lambda,\mu) = \begin{cases}
\partial^2 O_L(D,\theta) + \lambda^T \frac{\partial^2}{\partial D \partial \theta} (\theta) + \mu^T \frac{\partial^2}{\partial D \partial \theta} (\theta) = 0,
\end{cases}$$

where $\lambda^T_i (\theta) = 0, \mu^T_i (\theta) = 0, \mu^T_i (\theta) = 0$ and $\lambda^T_i (\theta) = 0$.

Bilevel Training-Set Attack Framework by Machine Teaching

**Bilevel Framework**

$$\min_{D \in \mathbb{D}, \theta} O_L(D,\theta)$$

s.t.

$$\begin{align*}
&\frac{\partial}{\partial \theta} O_L(D,\theta) + \lambda^T \frac{\partial^2}{\partial D \partial \theta} (\theta) + \mu^T \frac{\partial^2}{\partial D \partial \theta} (\theta) = 0, \\
&\lambda^T_i (\theta) = 0, \quad i = 1, \ldots, m, \\
&\mu^T_i (\theta) = 0, \quad i = 1, \ldots, m.
\end{align*}$$

**Upper-level: attacker**

Search space of feasible manipulations, e.g., data poisoned within budget.

**Lower-level: learner**

Overall attacker objective function, i.e.,

$$O_A(D,\widehat{\theta}_D) = R_A(\widehat{\theta}_D) + E_A(D,\widehat{\theta}_D).$$

**Attacker effort function, e.g.:**

$$E_A(D,\widehat{\theta}_D) = \|X - X_0\|^2_f$$

**Examples**

**SVM**

$O_L(D,\theta) = \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$

$\xi_i = 1 - \xi_i - y_i (x_i^T w + b)$

We convert it to the corresponding KKT conditions for $w_j$

$$\begin{align*}
0 &= \alpha_i - \alpha_i \sum_i I_i (1 - y_i (x_i^T w + b)) + \beta_i y_i x_i j = 0
\end{align*}$$

**The attacker** wants to make the learned weight close to the target weight $w^*$ by risk $R_A(\widehat{\theta}_D) = \|X - X_0\|^2_f$ and to minimally modify features by attacker's effort $E_A(D,\widehat{\theta}_D) = \|X - X_0\|^2_f$.

Combining them we get the KKT single-level framework

$$\begin{align*}
\min_{D \in \mathbb{D}, \theta} & \quad \frac{1}{2} \|w\|^2 + c \sum_i \xi_i \\
\text{s.t.} & \quad 0 = \alpha_i - \alpha_i \sum_i I_i (1 - y_i (x_i^T w + b)) + \beta_i y_i x_i j
\end{align*}$$

**Learning task:** given word frequencies in emails, the learner should classify them as spam/not spam.

**Target weight:** the attacker wants to make the weight on feature "credit" close to zero with minimal change of other weights. So we set feature "credit" frequency in training data to zero and refer the learned weight as the target weight.

**Logistic Regression**

$O_A(D,\theta) = \sum_i \log (1 + \exp (-y_i \theta_i x_i)) + \frac{1}{2} \|w\|^2_f$

**Learning task:** given features of wine the learner should classify good/bad wine.

**Target weight:** only correlated with feature "alcohol" (the 11-th feature).

**Attack behavior** is mainly increasing/decreasing the 11-th feature for good/bad data.