Poisoning Attacks against Support Vector Machines

ICML読み会 2012/07/28

Hidekazu Oiwa (@kisa12012)

oiwa (at) r.dl.itc.u-tokyo.ac.jp

読む論文

- Poisoning Attacks against Support Vector Machines
  - Battista Biggio (Itary), Blaine Nelson, Pavel Laskov (German)

    http://icml.cc/2012/papers/880.pdf (論文)
    - http://www.slideshare.net/pragroup/battista-biggio-icml2012-poisoning-attacks-against-support-vector-machines (スライド)
- 第一著者がDr. Laskovの元へ約半年visitingしてた時の論文
  - Adversarial Classificationの研究者

目次

- 研究の概要
  - Poisoning Attacksとは？
  - 問題設定
- 提案アルゴリズム
  - Poisoning Attacks against SVMs
  - カーネルSVMへの拡張
- 実験
  - 人工データ実験
  - 手書き文字認識実験

研究概要

背景

- （大規模）機械学習流行中
  - Malicious Behavior：悪意あるエージェントの行動
  - 分類器や異常検知器を混乱させる様に動く
    - Ex. スパムフィルタリング・マルウェア解析
- 目標：Malicious Behaviorに頑健なアルゴリズム
  - そのためには…
  - Malicious Behaviorの性質の分析が求められる

Malicious Behaviorの分類

[Barreno+ ML10]

- Causative Attack
  - 設計者が持っている訓練データを直接操作，書き換え
- Exploratory Attack
  - 設計者が持っている分類器を直接操作，書き換え
  - これらに対応したアルゴリズムは，すでに複数提案されている
- Poisoning Attack
  - 設計者の訓練データに新しく悪性データを注入
  - 他の手法と比べて，より現実的な攻撃方法
    - 設計者のデータベースを直接弄る必要がないため
  - 異常検知系の先行研究しかない [Kloft+ AISTATS10]+

Poisoning Attack

訓練データ

SVM

Poisoning Attack

訓練データ

SVM

Poisoning Attack

訓練データ

SVM

性能劣化

問題設定

設計者

training set

$\mathcal{D}_{tr} = \{x_i, y_i\}_{i=1}^{n}$

攻撃者

validation set

$\mathcal{D}_{val} = \{x_k, y_k\}_{k=1}^{m}$

悪性データ

(x_c, y_c)

SVMは悪性データを含めた
training setから学習

y_cは予め固定
Validation Setに対する分類性能を
一番押し下げるx_cを生成

本研究の概要

- SVMへのPoisoning Attacksの解析
  - 悪性データ作成アルゴリズムの提案
    - Incremental SVM
    - カーネル拡張
- 人工データ実験
- 手書き文字認識データ実験

提案アルゴリズム

最適化問題

$$\max_{x_c} L(x_c) = \sum_k [1 - y_k f_{x_c}(x_k)]_+ = \sum_k -g_k(x_c)$$

- Validation Setにおける損失を最大化
  - $f_{x_c}(\cdot)$: 悪性データ込みで学習されたSVM
  - 非凸な最適化問題
- 解法：Gradient Ascent
  - SVMの更新と悪性データの更新を繰り返す
  - ステップ幅を適切に設定すれば局所最適解に収束

$$x'_c = x_c + t \cdot u$$
$$u \propto \nabla L(x_c)$$

アルゴリズムの概要

Algorithm 1 Poisoning attack against SVM

Input: $\mathcal{D}_{\rm tr}$, the training data; $\mathcal{D}_{\rm val}$, the validation data; y_c, the class label of the attack point; $x_c^{(0)}$, the initial attack point; t, the step size.
Output: x_c, the final attack point.

1: $\{\alpha_i, b\} \leftarrow$ learn an SVM on $\mathcal{D}_{\rm tr}$.
2: $k \leftarrow 0$.
3: **repeat**
4: Re-compute the SVM solution on $\mathcal{D}_{\rm tr} \cup \{x_c^{(p)}, y_c\}$ using incremental SVM (*e.g.*, Cauwenberghs & Poggio, 2001). This step requires $\{\alpha_i, b\}$.
5: Compute $\frac{\partial L}{\partial u}$ on $\mathcal{D}_{\rm val}$ according to Eq. (10).
6: Set u to a unit vector aligned with $\frac{\partial L}{\partial u}$.
7: $k \leftarrow k+1$ and $x_c^{(p)} \leftarrow x_c^{(p-1)} + tu$
8: **until** $L\left(x_c^{(p)}\right) - L\left(x_c^{(p-1)}\right) < \epsilon$
9: **return:** $x_c = x_c^{(p)}$

初期点は，既存データのラベルをflipして作成

SVMの更新

勾配を算出

悪性データの更新

from [Biggio+ 12]

SVMの更新

- Incremental SVM [Cauwenberghs+ NIPS00]
  - １つずつデータを追加しながらSVMを学習
  - 全データの役割が不変な範囲の最適化問題を反復的に解く
    - Reserve Point / Support Vector / Error Vector

- 条件を破らないと収束しない場合，データの役割を変更
- 各最適化時には，サポートベクターのパラメータのみが更新
- データが追加されるたびに，全パラメータが収束するまで最適化すれば，SVMの最適解に収束

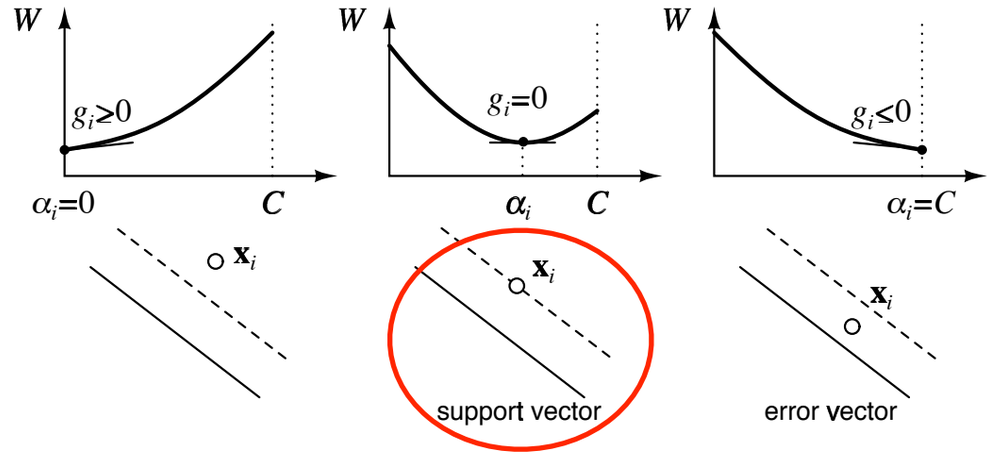


Figure 1: Soft-margin classification SVM training.
from [Cauwenberghs+ NIPS00]

最適化問題の勾配計算

- Incremental SVMのアイデアを用いる
  - 更新時に各データの役割が変動しない仮定を置く
  - サポートベクターのみに着目すれば良い
  - 更新式はカーネル関数に依存
- 厳密な計算には，条件を破らないステップ幅の導出が必要
  - 本研究では定数ステップ幅で値を更新，計算をサボる

$$\frac{\partial L}{\partial u} = \sum_{k=1}^{m} \left\{ M_k \frac{\partial Q_{sc}}{\partial u} + \frac{\partial Q_{kc}}{\partial u} \right\} \alpha_c, \qquad (10)$$

where

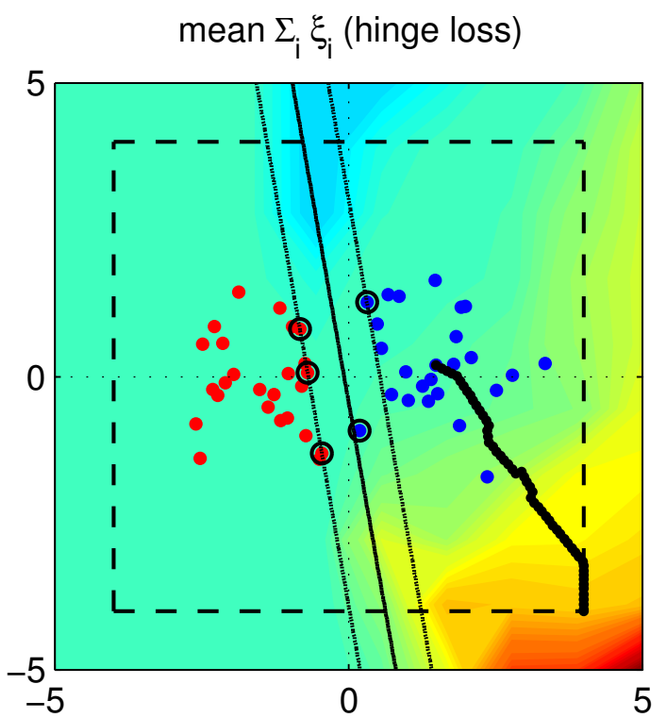
$$M_k = -\frac{1}{\zeta}(Q_{ks}(Q_{ss}^{-1} - \upsilon\upsilon^T) + y_k\upsilon^T).$$

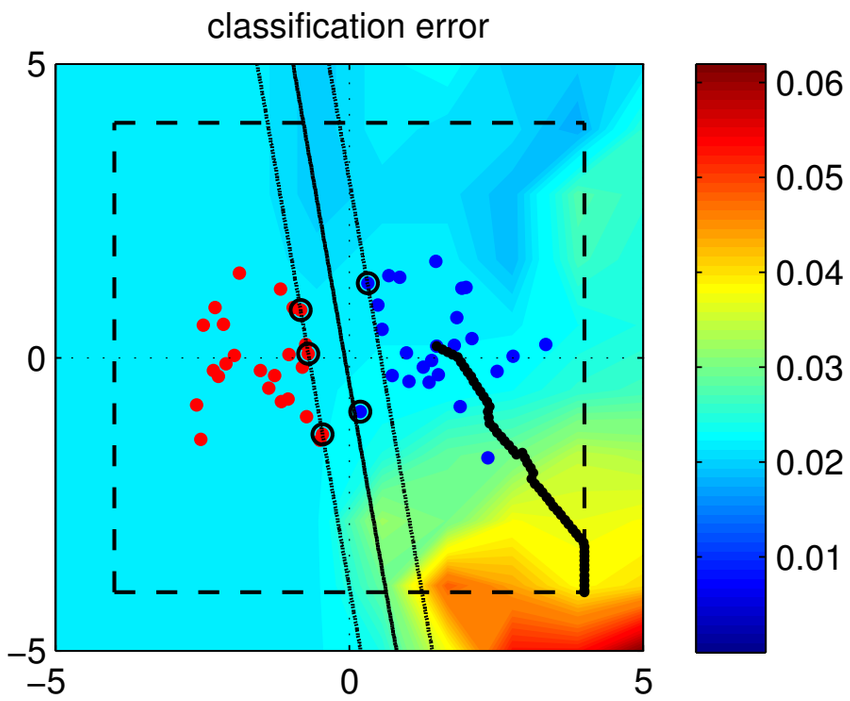
from [Biggio+ 12]

実験

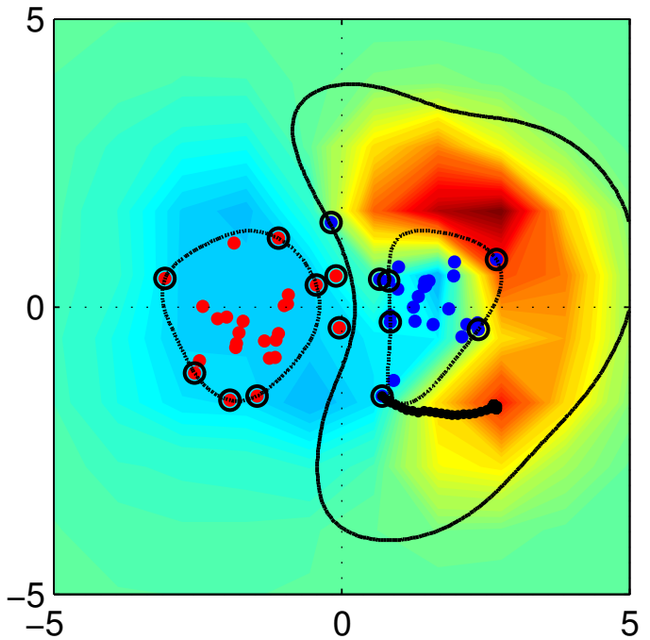
人工データ実験

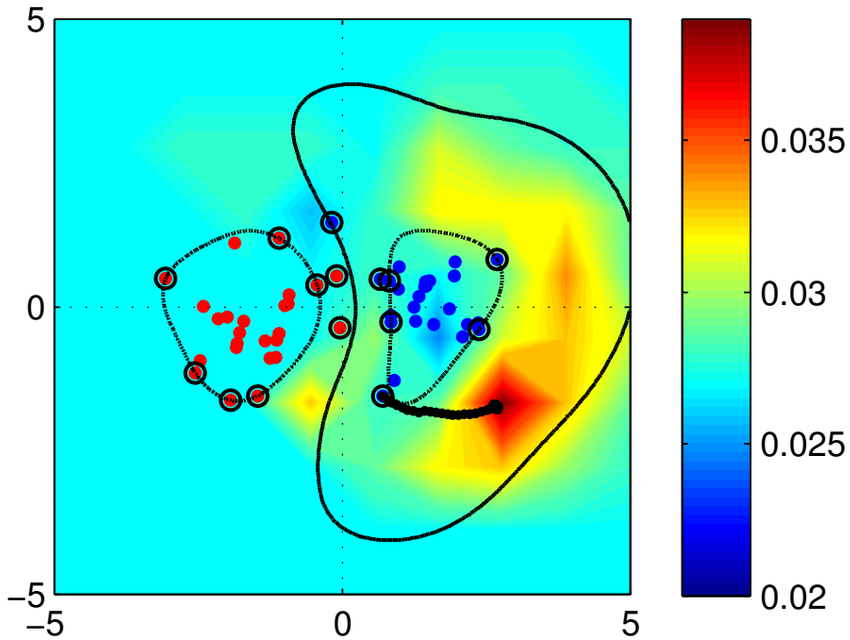
線形
カーネル





RBF
カーネル





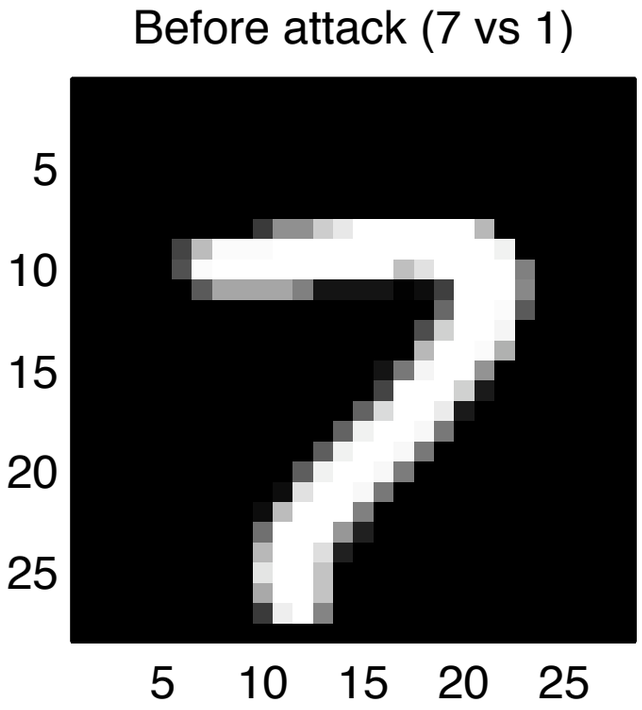
from [Biggio+ 12]

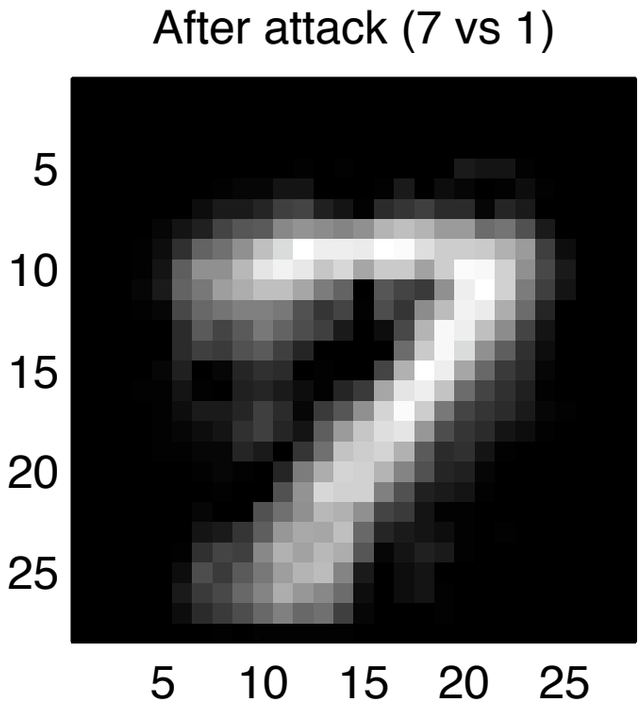
手書き文字認識

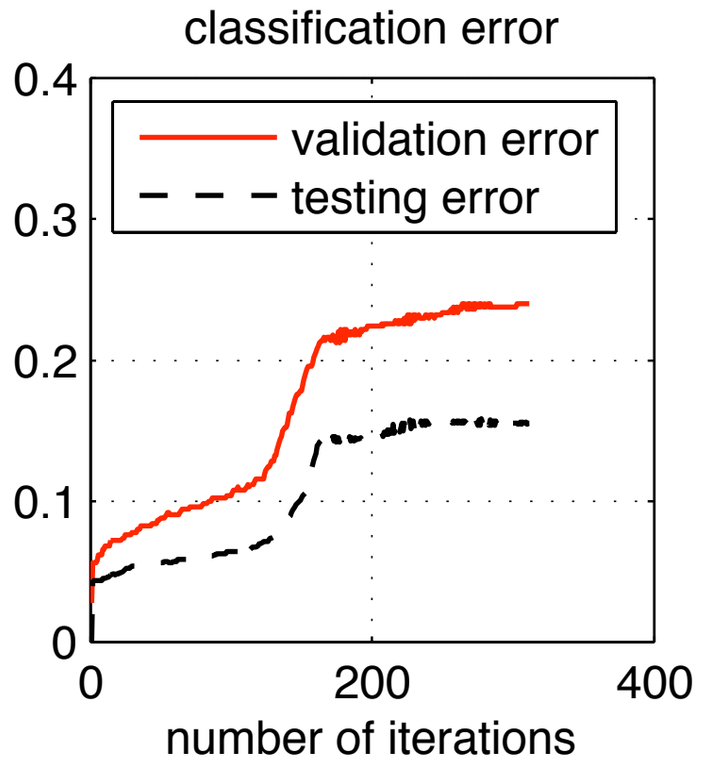
実験設定

実験データ	MNIST (7 vs. 1; 9 vs. 8; 4 vs. 0)
SVM	線形カーネル C=1
training set	100
validation set	500

手書き文字認識
実験結果 (7 vs. 1)



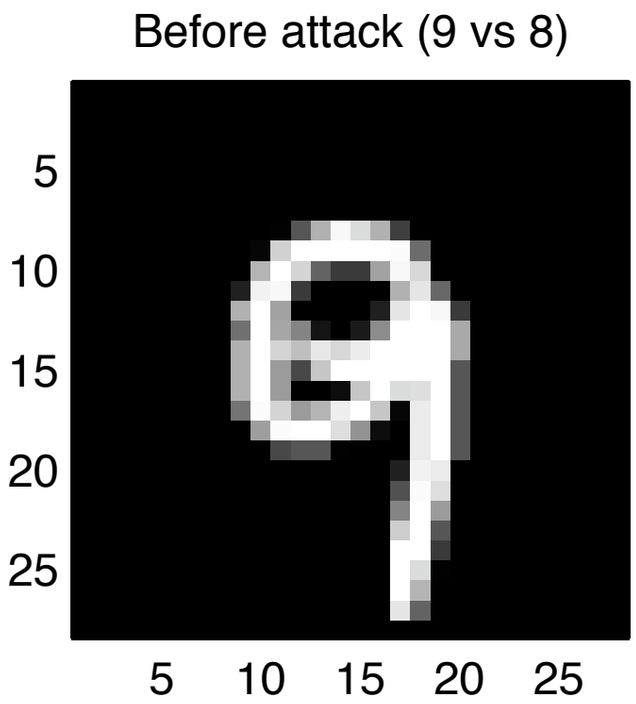


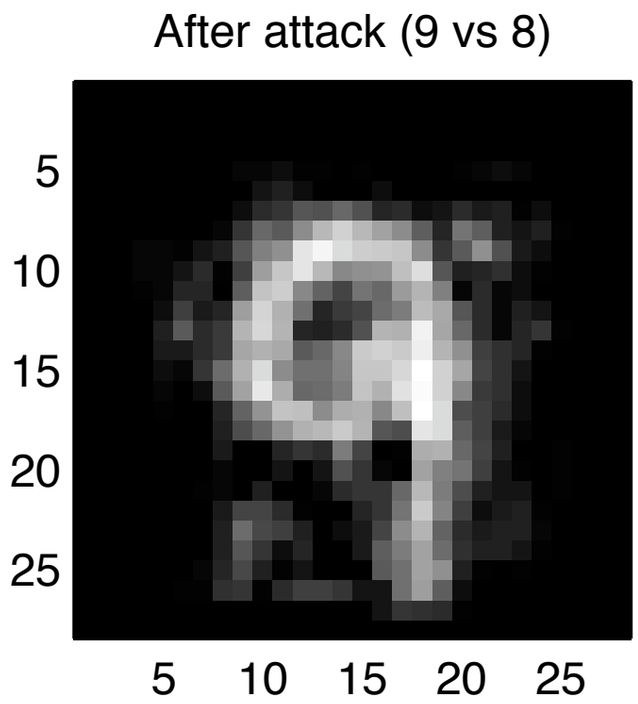


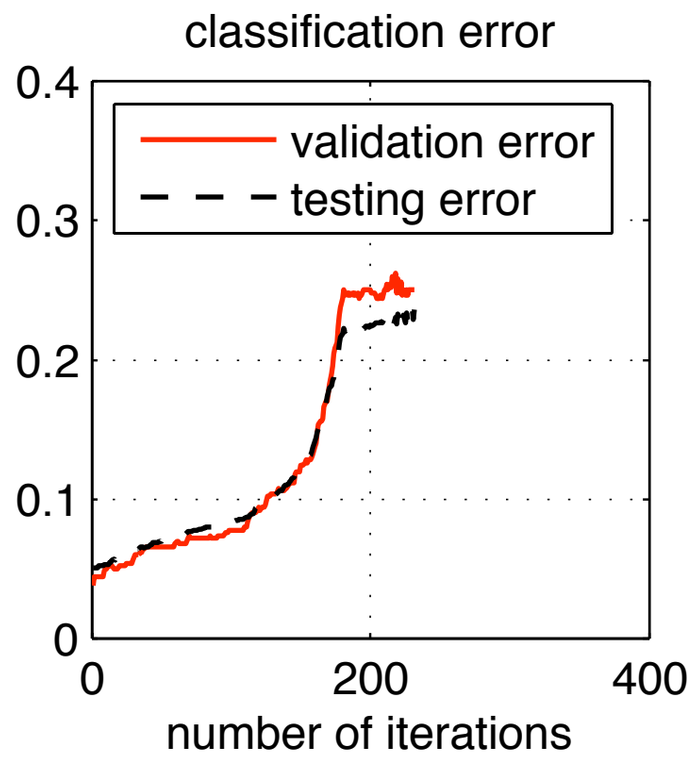
ラベルは1

from [Biggio+ 12]

手書き文字認識
実験結果 (8 vs. 9)



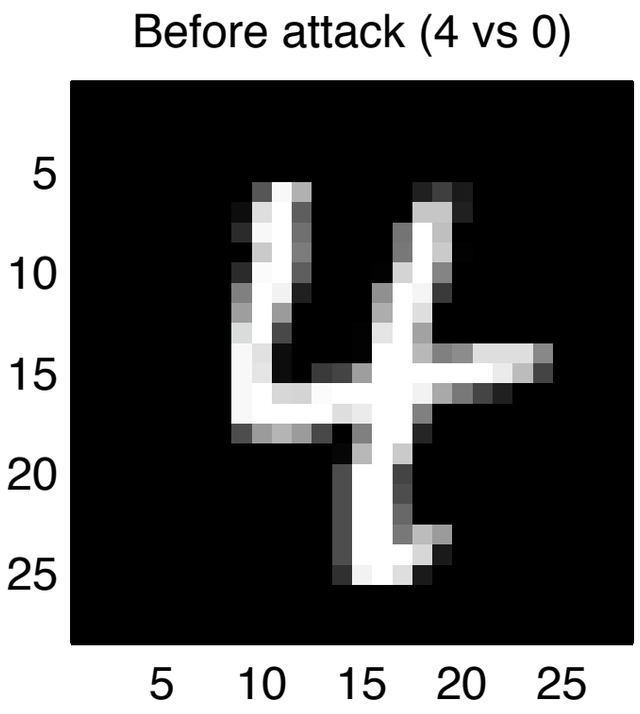


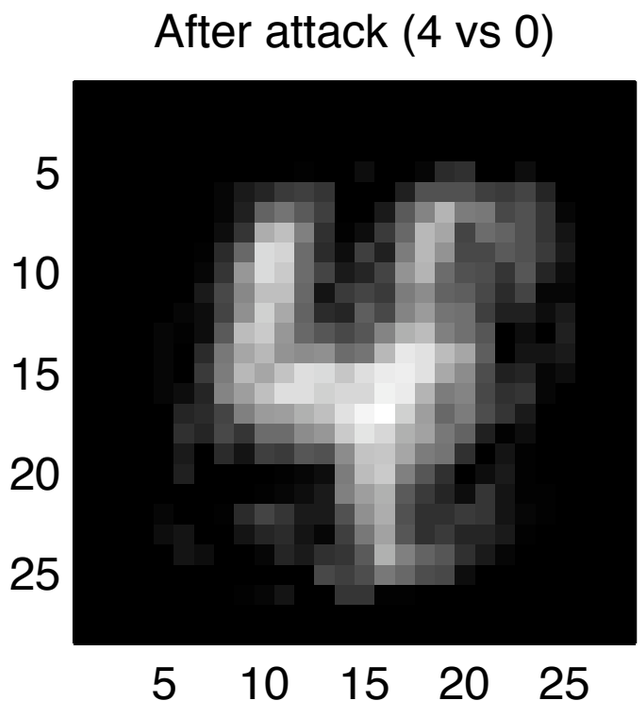


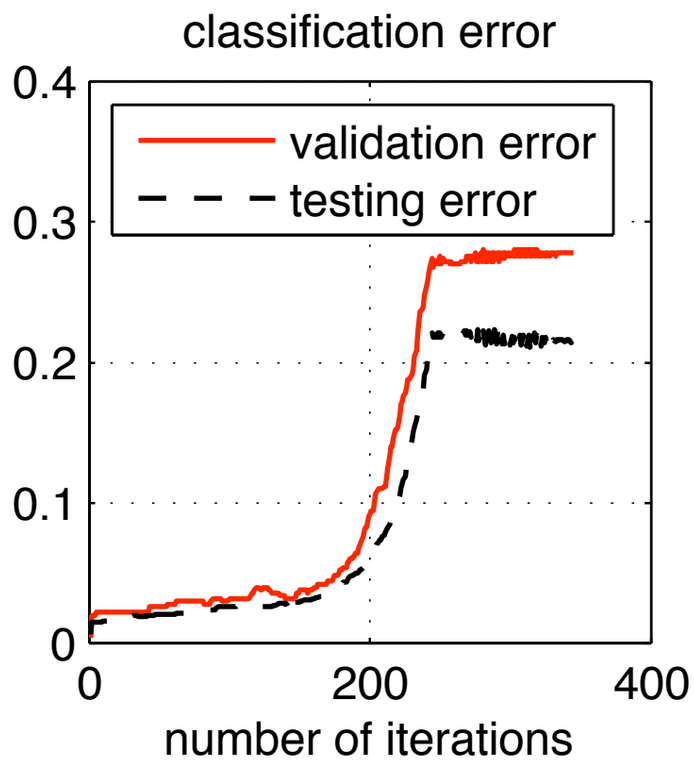
ラベルは8

from [Biggio+ 12]

手書き文字認識
実験結果 (4 vs. 0)







from [Biggio+ 12]

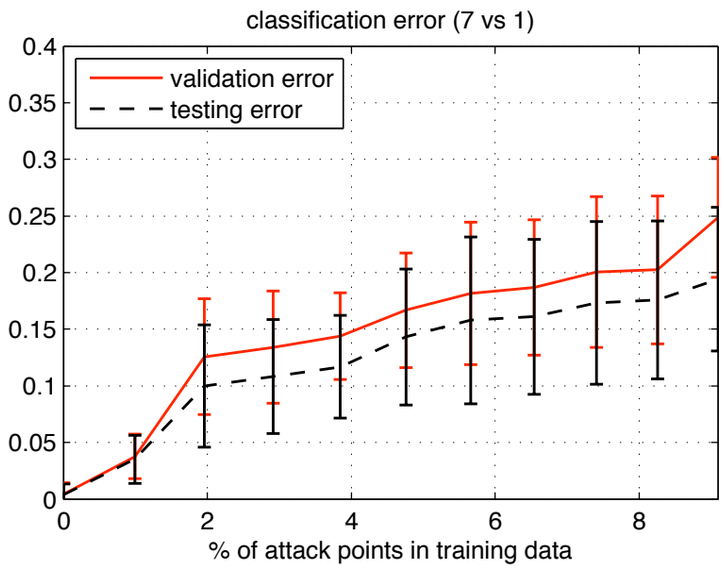
ラベルは0

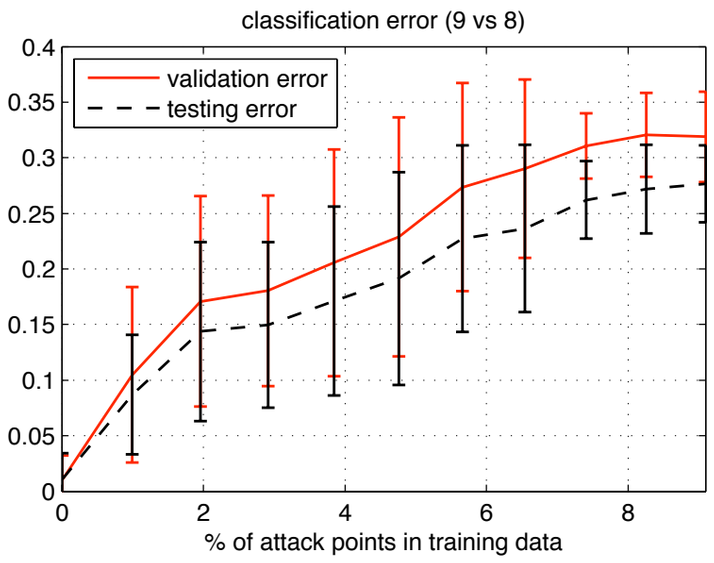
実験結果から

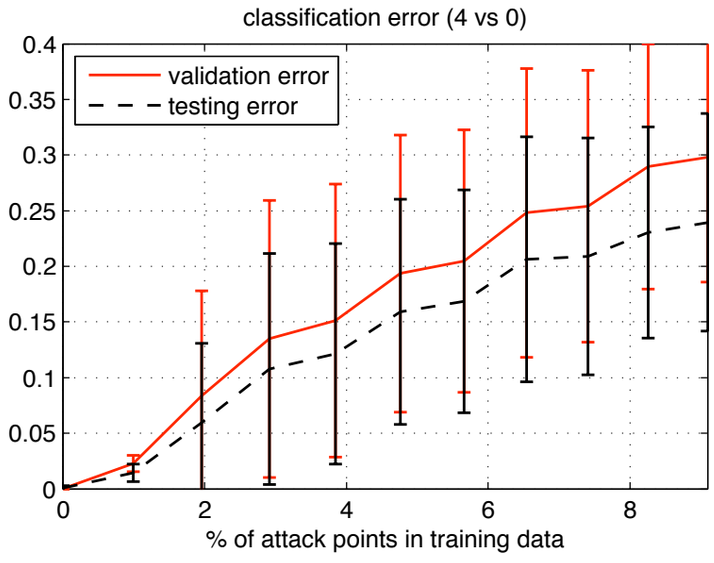
- 悪性データがラベルクラスの性質を取り込んだデータに変化
  - 7の下部が1の横棒に似た形になるなど
- 1データで，15-20%のエラー率向上を達成
  - 初期点では2-5%
  - その後のデータ改良にて，上記のエラー率を達成
  - アルゴリズムの有効性を示した形に
  - training dataの数を増やした時は，もっと性能は悪いでしょうが…

複数データ実験

- 悪性データを一個ずつ追加で入れていった場合の性能推移
  - 決定境界に近いデータを初期点に置くと，悪性データがreserve pointに陥るため，そこで更新がストップ







from [Biggio+ 12]

まとめ

- SVMに毒を盛る！
  - SVMの精度をガタ落ちさせるデータを新しく注入
- 性能を落とすデータの作り方を提案
  - 最適化問題は非凸だが，勾配法で解く形に
  - カーネルSVMにも適用可能
- 手書き文字認識タスクで実験
  - たった1データで精度を2割前後落とす事に成功

Future Work

- より効率的，頑健，高速な最適化手法
- カーネル毎でのPoisoning Attackへの耐性評価
- 複数の悪性データを同時に注入可能なケース
- データのラベルを攻撃者が固定できないケース
  - 設計者の人力でラベル付けされる場合等
    - ラベル付を誘導するため，入力データに制約が必要
- 現実的には，入力データの人工生成は困難
  - 入力ベクトルがbag-of-wordsの場合，それっぽいテキストに変換可能な結果を返す必要がある