Evasion attacks against machine learning at test time

年份：2013

1 Introduction

- 传统的算法或者模型的评估方法对于安全性能的评估是不合适的；
- 本文所提出的方法对于任何以*求导*为基础的算法或模型都是有效的；

2 Optimal evasion at test time

- 基本假设：$f : X \rightarrow Y$是一个区分样本是否合法（legitimate）的分类器，取值空间只有$\{-1, +1\}$，分别代表合法/不合法两类。D是在一个遵循概率分布为p(X, Y)的数据中采样得到的数据集。yc = f(x)代表分类器给出的结果，$g : X \rightarrow R$是一个连续函数，f的分类结果其实就是给g的输出划定门槛而得到的。在这之中我们假定g(x)如果小于0，则f为-1，反之亦然。

2.1 Adversary model

- Adversary's goal：攻击器的目标应该被定义为一个损失函数，而攻击器就是为了最大化（最小化）该损失函数。攻击器最好的策略应该是让分类器对攻击样本给出错误的分类，并给出很高的confidence。
- Adversary's knowledge：对于需要攻击的系统，攻击器所掌握的知识也是不确定的，包括目标系统训练时使用的训练集、真实样本的特征表示、算法类型或者模型的损失函数、目标模型分类器、分类器的feedback等等。
- Adversary's capability：在evasion攻击中，攻击器只允许修改测试数据，不允许修改训练数据。一般来说，攻击器可以直接修改input数据，也可以修改feature vectors、或者对特定的feature进行修改。

2.2 Attack scenario

- 如果我们有被攻击对象的完整信息，那么就可以直接以最小化g(x)作为目标。
- 但是如果没有被攻击对象的完整信息，只能知道部分信息，那么就通过构造分布相似的数据集，想方设法去重建一个相似的模型，再针对这个模型进行攻击。

2.3 Attack strategy

- 形式化定义evasion attack的过程，即为：

$$\begin{aligned}\mathbf{x}^* = \arg\min_{\mathbf{x}} \quad & \hat{g}(\mathbf{x}) \\ \text{s.t.} \quad & d(\mathbf{x}, \mathbf{x}^0) \leq d_{\max}.\end{aligned}$$

  其中x0是需要攻击的原始样本，x是修改后的攻击样本，x*是所有攻击样本中最优的那个。

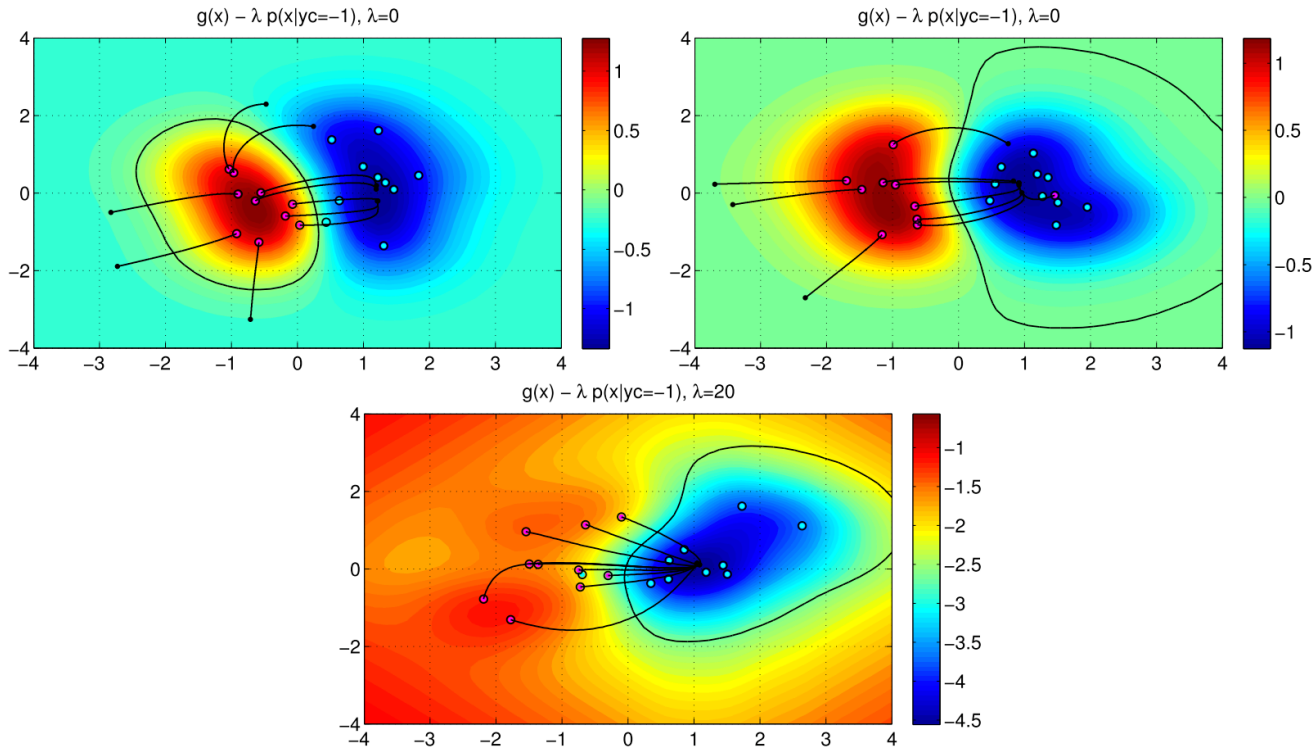
- 然而由于$\hat{g}(x)$只是一个对$g(x)$的逼近，且原始的$g(x)$并没有包含训练数据分布的相关信息，因此我们所得到的x^*可能不在被攻击对象的样本范围内（即$p(x) = 0$），因此这样的攻击对于被攻击对象来说可能是无效的。

- 因此，需要添加一个约束项让攻击样本尽量出现在原始数据分布较多的区域，所以新的目标函数整理为：

$$\arg\min_{x} \; F(\mathbf{x}) = \hat{g}(\mathbf{x}) - \frac{\lambda}{n} \sum_{i|y_i^c=-1} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \tag{2}$$

$$\text{s.t.} \; d(\mathbf{x}, \mathbf{x}^0) \leq d_{\max} \;\; , \tag{3}$$

其中k是核密度估计中的核函数，h是带宽。修正后的结果与原来的结果比较如下：



其中上面两幅图是没有添加约束项的结果，下面是添加了约束项的结果。颜色较深（红黄橙)的是攻击样本，颜色较浅的是原始合法样本，黑色曲线代表攻击样本逐渐收敛至最优解的过程。可以发现只有添加了约束项后，攻击样本才能基本朝着原始样本聚集的区域下降。

3 Gradients descent attacks

此处我们假设的是$g(x)$在任何地方都可导。如果某些地方不可导或者不光滑，那么也可以使用上述方法作为启发式的算法，具体工作可留给后人。

3.1 Gradients of discriminant functions

这一部分主要是回顾了常用分类器中判别函数的导数，如线性分类器、支持向量机、神经网络等。

3.2 Gradients of kernel density estimators

上述提到的KDE的导数与核函数k有关，文中主要考虑的是RBF核，即$k(\frac{x-x_i}{h}) = exp(-\frac{d(x-x_i)}{h})$，其中距离d一般常使用$l_1$或者$l_2$距离，则它们分别的导数为：

$$-\frac{2}{nh}\sum_{i|y_i^c=-1}\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|_2^2}{h}\right)(\mathbf{x}-\mathbf{x}_i)\ ,$$

$$-\frac{1}{nh}\sum_{i|y_i^c=-1}\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|_1}{h}\right)(\mathbf{x}-\mathbf{x}_i)\ .$$

3.3 Descent in discrete spaces

如果特征空间是离散的，那么$F(x)$梯度下降就有可能得到取不到的值，我们需要找到能够使$F(x)$下降最多的x。有两种方法，第一种是对x周围一定邻域内的所有可取值进行计算，取其中最小的一个；第二种是选择与$\nabla F(x)$同方向的值，虽然不一定是最小值，但是保证是下降的。

4 Experiments

4.1 A toy example on handwritten digits

作者使用MNIST数据集的28*28的手写数字照片作为原始样本，并将每个像素点标准化至[0,1]，d使用曼哈顿距离（即l_1），使用$d_{max} = \frac{5000}{255}$。被攻击的分类器是使用线性核的SVM，其中C=1。实验结果如下：

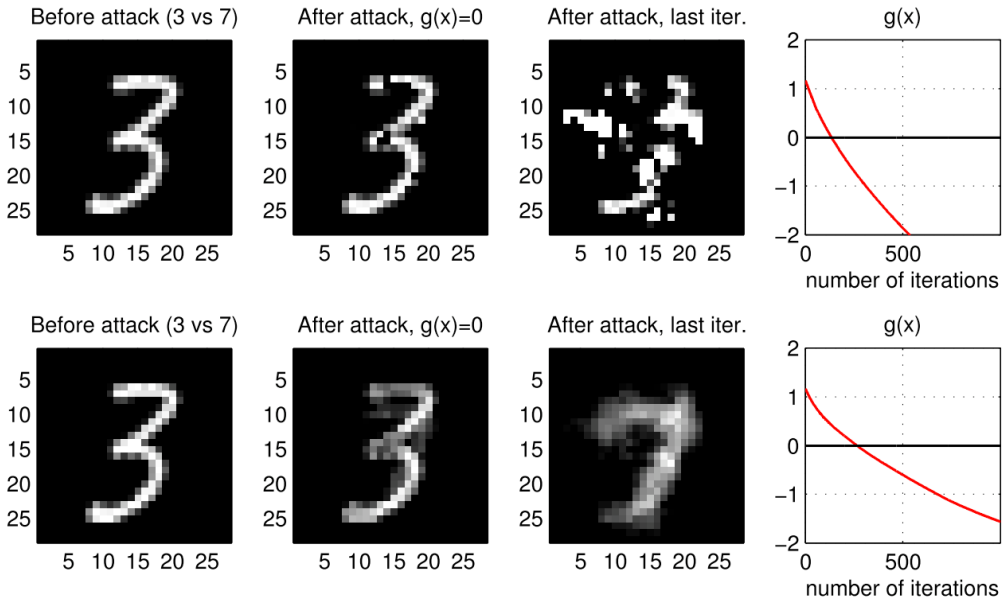


Fig. 3. Illustration of the gradient attack on the digit data, for $\lambda = 0$ (**top row**) and $\lambda = 10$ (**bottom row**). Without a mimicry component ($\lambda = 0$) gradient descent quickly decreases g but the resulting attack image does not resemble a "7". In contrast, the attack minimizes g slower when mimicry is applied ($\lambda = 0$) but the final attack image closely resembles a mixture between "3" and "7", as the term "mimicry" suggests.

在该例中，是否添加mimicry部分都能够达到攻击效果，但是添加了mimicry后显然攻击样本更趋向于"7"，而未添加mimicry的攻击样本则偏离比较远。

4.2 Malware detection in PDF file

在利用PDF进行恶意攻击时，对原始PDF进行内容添加较为简便，因此在原始问题的基础上又添加了约束$x^0 \leq x$，且d_{max}被定义为攻击者可以对PDF添加的keywords的最大数量。

实验结果如下：

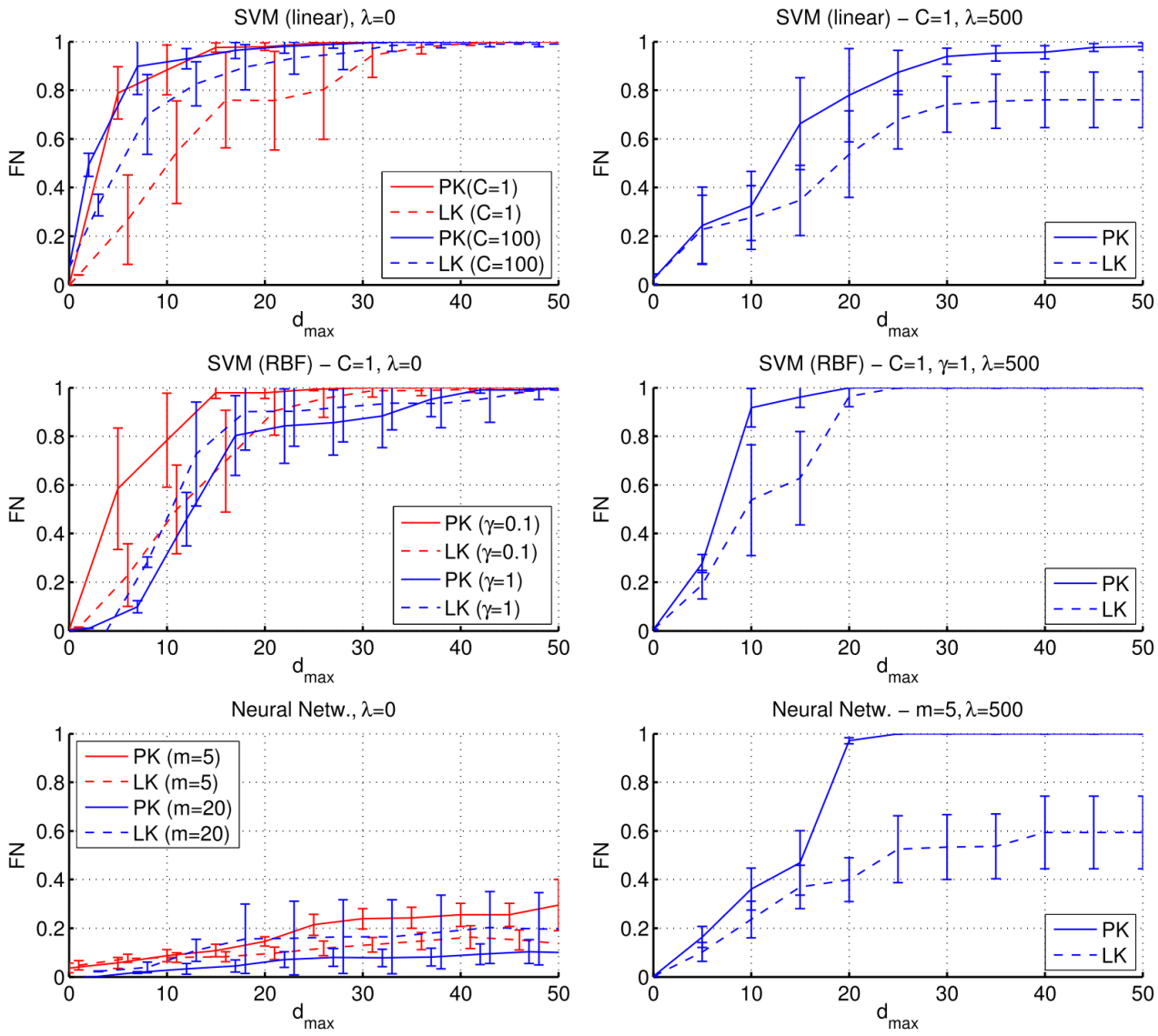


Fig. 4. Experimental results for SVMs with linear and RBF kernel (first and second row), and for neural networks (third row). We report the FN values (attained at FP=0.5%) for increasing $d_{\max}$. For the sake of readability, we report the average FN value ± half standard deviation (shown with error bars). Results for perfect (PK) and limited (LK) knowledge attacks with $\lambda = 0$ (without mimicry) are shown in the first column, while results with $\lambda = 500$ (with mimicry) are shown in the second column. In each plot we considered different values of the classifier parameters, *i.e.*, the regularization parameter C for the linear SVM, the kernel parameter γ for the SVM with RBF kernel, and the number of neurons m in the hidden layer for the neural network, as reported in the plot title and legend.

图中横坐标为修改的次数，纵坐标为当FPR为0.5时FNR的取值，能够反映当样本阳性时（即样本为攻击样本时），模型能否给出正确的判断，因此FNR就可以视为模型对攻击样本的鉴别错误率。显然，一个比较安全的分类器应该是FNR较低的分类器。

分析之后大致能得出，添加了约束项之后，对于线性分类器来说其实效果是下降了的，因为线性SVM本身就能够找到全局最低点，添加约束项使得它的梯度下降幅度降低，所花的时间也就更久。但是对于非线性分类器（比如神经网络）来说，添加了约束能够尽量让攻击样本朝着原始样本聚集的区域下降，保证被攻击的分类器训练时接触过类似的样本，从而不至于落入局部最低点，或者落入原始样本不曾触及到的特征空间。

5 Conclusions, limitations and future work

本文主要是提出了一种针对于可导分类器的攻击算法，主要通过添加约束项来提高攻击的成功率。作者认为这种算法可以被推广到不可导分类器的攻击中（如随机森林，决策树），只不过需要提出启发式的$g(x)$。同时，本文的研究也可以帮助分类器提高安全性，使用正则项将合法样本的边界缩小能够提高分类器的安全性，因为攻击样本需要“模仿”该范围内的合法样本才能完成攻击。作者也提到，一般来说，安全性的提高一般伴随着的都是更高的FPR。最后，作者提到也许使用集成方法（ensemble）或者bagging方法能够提升安全性。