Отравляющие атаки против SVM

Battista Biggio, Blaine Nelson, Pavel Laskov

28.02.2024

Abstract

Мы исследовали семейство отравляющих атак против Метода опороных векторов(SVM). Эо атаки вводятся специально в созданной трейн, чтобы увеличить тестовую ошибку SVM. Ключевая мотивация для этих атак - это факт того, что большинство обучащих алгоритов предполагают что их трейн получен из натурального или грамотно-подобрного распределения. Однако, это предположение обычео не выполняется в установках чувстительных к безопасности. Как мы покажем, аккуратное влияние может, в какой то степени, предсказать изменение предсказательной функции из-за вредоносного вода и использовать это возможность для создания вредоносной даты. Предложеная атака использует стратегию градиентного подьеам в который градиент вычисляется на харатеристиках оптимального решеня SVM. Этот метод может быть введен в ядро и позвоялет сконструировать атаку во входящем просранстве даже в не-линеном ядре. Мы экспериментально показали, что наша процедура увеличения градиента с большой вероятностью находит зороший локальный максимум невыпуклой поверхности валидационной ошибки, которая существуенно увеличивает тестовую ошибку классификатора.

1 Введение

ML техники стремительно появились как ждый инструмент в разнообразии онлайн больших систем приложений, потому что они могут выявлять спрятанные закономерности в больших сложных наборах данных, адаптируясь к новому поведению, и обеспечивать статистическую обоснованность для процесса принятия решений. Разработчики приложений могут использовать обучение таким образом, чтобы помогать решать задачи, который называются big-data problembs и они включает число связанных с безопасностью задач, в частности фокусируясь на нахождении вредоносного или нетепичного поведения. Фактически, подходы обучения уже были использованы или предлогалаись как решения задач, связанных с безопасноть., включая спам, червей вторжения и мошенничества. К несчастью, в этих областях, дата это обычно не только нестанционарное но и также имеет уязвимый компонент, и гибкость предоставленния техникам обучения может использована атакующих для достижения его целей. Например, в спам-детекции, нападающие обычно адаптируют их подходы основаные на популярных спам детекторах, и часто умный нападающий изменит его поведение или увильнет, либо введет в заблуждение. В ответ на угрозу недобросовестного манипулирования данными несколько предложенных методов обучения явно учитывают определенные типы поврежденных данных. Атак против обучающих алгоритмов могут быть классифицированы, среди других категорий, в причинные (манипуляция над тренировочной информацией) и разведочный(иследования классификатора). Отравление относится к причинным атакам в который специально созданые атаки вводяться в тренировочный датасет. Это атаки особенно важны с практической точки зрения , так как атакующих обычно не имеет прямого доступа к существующему трейну, но может предложить новый обучащие данные, т. е. Открытые репозитории и соблазн часто собирают вредоносные примеры для обучения, которые предоставляют возможность нападающему для отравления тренировочной даты. Отравляющие атаки могут быть раньше были изучеены для для простых методов детекции аномалий. В этой статье, мы проверим семейтво отравляющих атак против SVM. Следуя обычному методу анализа защиту для ML, мы допустим, что атакующий знает алгоритм обучения и может взять информацию из основного распределения информации. Далее, мы предположим, что наш атакующий знает трейн испольщующися обучаемым, обычно, в нереалистичном допущении, но в реальном мире, злоумышленник мог бы вместо этого использовать суррогатный обучающий набор, взятый из того же дистрибутива и наш подход анализ возможностей злоумышленника в худшем случае. Под этими упущениями, мы презентуем новый метод, который атакующий может использовать для создания точки данных, которая существенно уменьштает очность SVM классификатора. Предложенный метод основан на свойствах оптимального решения задачи SVM обучения. Как было впервые показано в методике поэтапного обучения, это решение плано зависит от параметров задачи квадратичного программирования и геометрии дата точек. Мы покажем, что такие найдены точки могут быть сформулирована как оптиимзазция по показателю производительности, при усвлоии сохранения оптимального задачи обучения SVM. Хотя поверхность тестируемой обычно невыпуклая, процедура подьема по градиенту в нашем методу надежно назеодет локальный максимум поверхности тестируемой ошибки. Данный предложенный метод зависит только от градиентов скалярныхх произведений между точками во входном пространстве, и следовательно, можт быть ядризован. Это отличается от предыдущей работы, свзяанной с созданием специальных точек атаки, в которых атака может только быть сконструирована в пространстве признаков для нелинейного случая. Последнее это сильное препятствие для атакующего, так как он должуен конструировать дату во входном простнатсве и не имеет практического средства доступа к пространству признаков. Следовательно, предложенный метод ломает новую почву в улучшении влияния основанных на данных атак против ядерных обучающих алгоритмов и подчеркивает необходимость рассмотрения сопротивления против негативной обучающей даты как важдый фактор конструирования обучащих алгоритмов.

2 Отравляющие атаки на SVM

Мы возьмем, что SVM был обучен на датасете $D_{tr} = \{x_y, y_i\}_{i=1}^n$, $x_i \in R^d$. Следуя стандартному обозначению, K обозначает матрица ядерных значений между двумя множествами точек, $Q = yy^T \odot K$ означает К снабженную метками, и α означает SVM двойным метка относящимся к каждой точке. В зависимости от значения α_i, обучающие точки относятся к margin support vectors($0 < \alpha_i < C, setS$, error support vector ($\alpha_i = C, setE$) и запасные точки ($\alpha_i = 0, setR$). В этой последовательности нижние регистры s,e,r используются для индексации к относящимся частям векторов или матриц; т.е. Q_{ss} означает margin support vector подматрицу Q.

2.1 Main Derivation

В отравляющих атак, цель атакующего найти точку (x_c, y_c), чье добавление к D_{tr} максимально уменьшает точность SVM классификатора. Выбор для атаки точки метки y_c произвольный но фиксированный. Мы относим эту выбранную метку к атакующему классу и другие относим к атакованному классу.

Атакующий продолжает, создавая валидный датасет $D_{val} = \{x_k, y_k\}_{k=1}^m$и максимизируя полученнуый ушерб понесенный благодаря D_{val} во время тренировки SVM на $D_{tr} \cup (x_c, y_c)$:

$$\max_{x_c} L(x_c) = \sum_{k=1}^{m} (1 - y_k f_{x_c}(x_k))_+ = \sum k = 1^m (-g_t k)_+ (1)$$

Здесь, мы берем роль атакующего и разрабатыаем метод для оптимизации x_c с этой целью. Первое,мы явно посчитаем для всех слагаемых в маржинальном уравнении g_k , которая пострадала из-за x_c:

$$g_k = \sum_j Q_{kj}\alpha_j + y_k b - 1 = \sum_{j!=c} Q_{kj}\alpha_j(x_c) + Q_{kc}(x_c)\alpha_c(x_c) + y_k b(x_c) - 1(2)$$

Несложно увидеть с этих уравнений, что $L(x_c)$ это невыпуклая целевая функция. Таким образом, мы используем технику увеличения градиента чтобы итеративно оптимизировать его. Мы берем что взятая область атаакующей точки $x_c^{(0)}$ была выбрана. Наша цель обновить атакующую точку следующим образом : $x_c^p = x_c^{p-1} + tu$ где p - текущая итерация, u - нормальный единичный вектор обозначающий направление атаки и t размер шага. Определенно, чтобы максимизировать наш обьект, атакующее направление u выравнивает градиент L с учетом u, который должен быть вычислен для каждой итерации. Однако hinge loss ни везде дифференцируема, это можно преодолеть только добавляем точки индексом k и ненулевым значением для L, т.е. для которого $-g_k > 0$. Влияние этих точек к гранинду L может быть вычислен благодаря дифференцированию 2 уравнения с учетом u используя правило произведения:

$$\frac{\partial g_k}{\partial u} = Q_{ks}\frac{\partial \alpha}{\partial u} + \frac{\partial Q_{kc}}{\partial u}\alpha_c + y_k\frac{\partial b}{\partial u}(3)$$

, where

$$\frac{\partial \alpha}{\partial u} = \begin{bmatrix} \frac{\partial \alpha_1}{\partial u_1} & \cdots & \frac{\partial \alpha_1}{\partial u_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial \alpha_s}{\partial u_1} & \cdots & \frac{\partial \alpha_s}{\partial u_d} \end{bmatrix} (3) simil. \frac{\partial Q_{kc}}{\partial u}, \frac{\partial b}{\partial u}.$$

Выражение для градиента далее может быть улучшен используя тот факт, что шаг взять в направлении u может сохранять оптимальное SVM решение. Это может быть пресдавлено как адиабатическое обновление состояния используя технику в этой статье. Исследуя, что для i-ой точки в тренировочном датасете, KKT состояний для оптимального решения проблемы SVM обучения может быть выражена как:

$$g_i = \sum_{j \in D_{tr}} Q_{ij}\alpha_j + y_i b - 1$$

$$\begin{cases} > 0; & i \in R \\ = 0; & i \in S \ (4) \\ < 0; & i \in E \end{cases}$$

$$h = \sum_{j \in D_{tr}} y_j \alpha_j = 0 \ (5)$$

Неравенства 4 и 5 значат, что бексконечное маллое изменение в аттакующей точке x_c повод мягко в оптимальном решении SVM, под условием, что композиция множеств S,E и R остается нетронутым. Эта точка равновсия позволяет нам предсказать реакцию решения SVM на изменение x_c, как показано ниже. Деффиринцируя по каждой слагаемой зависимой по x_c с учетом каждого компонента u_l $(1 <= l <= d)$, мы получаем для каждого $i \in S$,

$$\frac{\partial g}{\partial u_l} = Q_{ss} + \frac{\partial Q_{sc}}{\partial u_l}\alpha_c + y_s \frac{\partial b}{\partial u_l} = 0$$

$$\frac{\partial h}{\partial u_l} = y_s^T \frac{\partial \alpha}{\partial u_l} = 0,$$

что может быть переписана как:

$$\begin{bmatrix} \frac{\partial b}{\partial u_l} \\ \frac{\partial \alpha}{\partial u_l} \end{bmatrix} = \begin{bmatrix} 0 & y_S^T \\ y_S & Q_s s \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \frac{\partial Q_{sc}}{\partial u_l} \end{bmatrix}^{-1} \alpha_c. \ (7)$$

Первая матрица может быть повернута с помощью формулы Шермана-Морисона-Вудбэри:

$$\begin{bmatrix} 0 & y_s^T \\ y_s & Q_{ss} \end{bmatrix}^{-1} = \frac{1}{\zeta} \begin{bmatrix} -1 & v^T \\ v & \zeta Q_{ss}^{-1} - vv^T \end{bmatrix} (8)$$

где $v = Q_{ss}^{-1} y_s \ and \ \zeta = y_s^T Q_{ss}^{-1} y_s$. Замена 8 в 7 и обнаружение того, что все компоненты повернутой матрицы независимы от x_c, вы увидим:

$$\frac{\partial \alpha}{\partial u} = \frac{-1}{\zeta}\alpha_c(\zeta Q_{ss}^{-1} - vv^T) \cdot \frac{\partial Q_{sc}}{\partial u}$$

$$\frac{\partial b}{\partial u} = \frac{-1}{\zeta}\alpha_c v^T \cdot \frac{\partial Q_{sc}}{\partial u}$$

Замена в 9 в 3 и далее в 1, мы получаем желаемый градиент, использующийся в оптимизации нашей атаки:

$$\frac{\partial L}{\partial u} = \sum_{k=1}^{m} \{M_k \frac{\partial Q_{sc}}{\partial u} + \frac{partialQ_{ks}}{\partial u}\}\alpha_c$$

где

$$M_k = \frac{-1}{\zeta}(Q_{ks}(\zeta Q_{ss}^{-1} - vv^T t) + y_k v^T)$$

2.2 Ядерность

Из уравнения 10, мы увидели что градиент целевой функции в k-ой итерации может зависеть от атакующей точки $x_c^{(p)} = x_c^{p-1} + tu$ только через градиенты матрицы Q. В частности, он распрстраняется на выбранное ядро. Мы ниже опишем выражение этих градиентов для трих стандартных ядер. Линейное ядро:

$$\frac{\partial K_{ic}}{\partial u} = \frac{\partial(x_i \cdot x_c^{(p)})}{\partial u} = tx_i$$

Полиномиальное ядро:

$$\frac{\partial K_{ic}}{\partial u} = \frac{\partial(x_i \cdot x_c^{(p)} + R)^d}{\partial u} = d(x_i \cdot x_c^{(p)} + R)^{d-1} tx_i$$

RBF ядро:

$$\frac{\partial K_{ic}}{\partial u} = \frac{\partial e^{\frac{-\gamma}{2}||x_y - x_c||^2}}{\partial u} = K(x_i, x_c^{(p)})\gamma t(x_i - x_c^{(p)})$$

Зависимость на $x_c^{(p)}$ (и ,конечно, на u) в градиентах нелинейных ядер может быть избежена заменой $x_c^{(p)}$ yf $x_c^{(p-1)}$, предусмотрим, что t досточно мало. Это приближение позволяет на простое расширение нашего метода на случайное ядро.

2.3 Алгоритм отравляющей атаки

Алгоритмические делати метода описаны в main derivation представляют Алгоритм 1.

Алгоритм 1. Отравляющая атака против SVM **Input** : D_{tr}, тренировочной датасет; D_{val} валидационный датасет; y_c, класс меток атакующей точки; x_c^0; t размер шага.

Output : x_c финальная атакующая точка.

1. $\{\alpha_i, b\} \leftarrow$ learn an SVM on D_{tr}

2. $k \leftarrow 0$.

3. **repeat**