

Отравляющие атаки против SVM

Battista Biggio, Blaine Nelson, Pavel Laskov

28.02.2024

Abstract

Мы исследовали семейство отравляющих атак против Метода опорных векторов (SVM). Эти атаки вводятся специально в созданной тренин, чтобы увеличить тестовую ошибку SVM. Ключевая мотивация для этих атак - это факт того, что большинство обучающих алгоритмов предполагают что их тренин получен из натурального или грамотно-подобного распределения. Однако, это предположение обычно не выполняется в установках чувствительных к безопасности. Как мы покажем, аккуратное влияние может, в какой то степени, предсказать изменение предсказательной функции из-за вредоносного ввода и использовать это возможность для создания вредоносной даты. Предложенная атака использует стратегию градиентного подъема в который градиент вычисляется на характеристиках оптимального решения SVM. Этот метод может быть введен в ядро и позволяет сконструировать атаку во входящем пространстве даже в не-линейном ядре. Мы экспериментально показали, что наша процедура увеличения градиента с большой вероятностью находит хороший локальный максимум невыпуклой поверхности валидационной ошибки, которая существенно увеличивает тестовую ошибку классификатора.

1 Введение

ML техники стремительно появились как ждкий инструмент в разнообразии онлайн больших систем приложений, потому что они могут выявлять скрытые 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 problems и они включают число связанных с безопасностью задач, в частности фокусируясь на нахождении вредоносного или нетипичного поведения. Фактически, подходы обучения уже были использованы или предлагались как решения задач, связанных с безопасностью, включая спам, червей вторжения и мошенничества. К несчастью, в этих областях, дата это обычно не только нестационарное но и также имеет уязвимый компонент, и гибкость предоставления техник обучения может использоваться атакующих для достижения его целей. Например, в спам-детекции, нападающие обычно

адаптируют их подходы основанные 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 (исследования классификатора). Отравление относится к причинным атакам в который специально созданные атаки вводятся в тренировочный датасет. Это атаки особенно важны с практической точки зрения, так как атакующий обычно не имеет прямого доступа к существующему трейну, но может предложить новый обучающие 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 изучены для для простых методов детекции аномалий. В этой статье, мы проверим семейство 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 использующийся 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 уменьшает 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 оптимизация по показателю производительности, при условии сохранения оптимального задачи обучения SVM. Хотя поверхность тестируемой обычно невыпуклая, процедура подъема по градиенту в нашем методу надежно назовет локальный максимум поверхности тестируемой ошибки. Данный предложенный метод зависит только от градиентов скалярных произведений между точками во входном пространстве, и следовательно, может быть ядризован. Это отличается от предыдущей работы, связанной 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 должен конструировать дату во входном пространстве 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 важный фактор конструирования обучающих 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.

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 полученный ущерб понесенный благодаря D_{val} во время тренировки SVM на $D_{tr} \cup (x_c, y_c)$:

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

Здесь, мы берем роль атакующего и разрабатываем 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 :

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

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 атакующей 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 объект, атакующее направление и 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

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

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

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

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

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

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

$$\begin{aligned} \frac{\partial g}{\partial u_l} &= Q_{ss} + \frac{\partial Q_{sc}}{\partial u_l} \alpha_c + y_s \frac{\partial b}{\partial u_l} = 0 \\ \frac{\partial h}{\partial u_l} &= y_s^T \frac{\partial \alpha}{\partial u_l} = 0, \end{aligned}$$

что может быть переписана как:

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

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

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

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:

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

где

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