Методы классификации из ML в ИБ

Введение

1. **Титульный слайд (название модуля, автор)**

Приветствие. Занятие «Методы классификации для информационной безопасности». Третий урок в модуле «Искусственный интеллект в информационной безопасности».

1. **Программа модуля**

На лекции: разница между методами машинного обучения (МО) с учителем и без, базовые алгоритмы классификации, применение алгоритмов классификации в информационной безопасности. На практическом занятии: анализ спама (классификация)

1. **Машинное обучение**

Привести примеры определений МО (на слайде). Артур Самуэль − изобретатель первой самообучающейся компьютерной программы игры в шашки. Том Митчелл - американский учёный, основатель первой в мире кафедры машинного обучения и автор первого учебника по этому предмету. «МО – это изучение компьютерных алгоритмов, которые могут автоматически улучшаться благодаря опыту и использованию данных».

1. **Машинное обучение: задачи и направления**

Методы МО делятся на методы без учителя и с учителем. Принципиально отличаются наличием (или отсутствием) «истинно-верных» ответов при обучении модели.

<https://docs.microsoft.com/ru-ru/dotnet/machine-learning/resources/tasks>

1. **Обучение с учителем: регрессия, классификация (общие подходы)**

Классификация – предсказание одного из конечного списка классов, регрессия – предсказание одного из бесконечного списка классов. Опирается на шаблоны данных, а не на явное программирование.

<https://habr.com/ru/company/ods/blog/322534/>

1. **Классификация: деревья принятия**

Дерево принятия решений — это двоичное дерево, каждый узел которого - входная переменная и точка разделения для этой переменной (если переменная — число). Листовые узлы — это выходная переменная, которая предсказывается. Классификация выполняется с помощью прохода по дереву к листовому узлу. Смысл обучения модели – получение условий и весов для узлов.

Модель умеет только интерполировать, но не экстраполировать. То есть дерево решений делает константный прогноз для объектов, находящихся в признаковом пространстве вне параллелепипеда, охватывающего все объекты обучающей выборки.

Преимущества:

* интерпретируемость: правила классификации понятны человеку
* быстрое обучение и предсказание

Недостатки:

* чувствительны к шумам
* может только интерполировать,  
  но не экстраполировать

Графическое представление работы – на слайде.

<https://habr.com/ru/company/ods/blog/322534/>

1. **Классификация: K-ближайших соседей**

Метрический метод классификации. Объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны

Преимущества:

* интерпретируемость: правила классификации понятны человеку
* устойчив к выбросам (голосование)

Недостатки:

* вычислительно сложный (используются все доступные данные)
* зависит от выбранной метрики расстояния
* нет теоретических оснований выбора определенного числа соседей (на практике берут нечётные)

Применять, когда много признаков (проклятие размерности).

Графическое представление работы – на слайде.

<https://habr.com/ru/company/ods/blog/322534/>

1. **Классификация: метод опорных векторов**

Пространство разделяется многомерными плоскостями, отделяющими области разных классов

Преимущества:

* хорошо работает в условиях большого количества признаков
* может обучаться на небольших данных

Недостатки:

* долго обучается на больших данных
* чувствителен к шуму
* нет общепринятого метода подбора ядер (для плоскостей сложной формы)

Графическое представление работы – на слайде.

<https://habr.com/ru/company/ods/blog/484148/>

1. **Классификация: байесовский классификатор**

Отдельные признаки рассматриваются как независимые друг от друга. С использование теоремы Байеса апостериорные вероятности описываются через априорные.

Преимущества:

* не требует подбора гиперпараметров
* работает быстро

Недостатки:

* не обнаруживает категории, отсутствовавшие при обучении
* предположение о независимости признаков

Пример применения показать на слайде (разобран случай определения вероятности исхода)

<https://labelme.medium.com/наивный-байесовский-классификатор-naive-bayes-classifier-b939578f6e>

<http://datareview.info/article/6-prostyih-shagov-dlya-osvoeniya-naivnogo-bayesovskogo-algoritma-s-primerom-koda-na-python/>

<https://linis.hse.ru/data/2016/03/23/1128107554/models-probability.pdf>

1. **Методы регрессии: линейная регрессия**

Модель зависимости непрерывной переменной от одной или нескольких других переменных с линейной функцией зависимости

Преимущества:

* Скорость и простота обучения
* Интерпретируемость: по коэффициентам можно понять влияние факторов

Недостатки:

* Чувствительна к выбросам
* Неэффективна при нелинейной зависимости целевой переменной от входной

Графическое представление работы – на слайде.

<https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/>

1. **Методы регрессии: логистическая регрессия**

Классификация, а не регрессия (несмотря на название)

Решает задачу предсказания значения непрерывной переменной, принимающей значения от 0 до 1: вероятность класса P=1/(1+𝑒^(𝑓(𝑏, 𝑋)) ), где X − признаки, b – коэффициенты

Преимущества:

* Скорость и простота обучения
* Интерпретируемость: по коэффициентам можно понять влияние факторов

Недостатки:

* Чувствительна к выбросам
* Неэффективна при нелинейной зависимости целевой переменной от входной

Графическое представление работы – на слайде.

<https://habr.com/ru/company/io/blog/265007/>

1. **Ансамблирование моделей**

Обучаются несколько базовых моделей, затем их результаты объединяются по какому-либо правилу и формулируется окончательный результат

Если использование простой модели машинного обучения не даёт достаточно качественного результата, то можно объединить несколько моделей. Ткой метод называется ансамблированием моделей. Основные варианты ансамблирования – бэггинг, бустинг, стекинг.

Бэггинг – это способ объединения нескольких моделей, при котором из исходных данных формируются выборки, каждая из которых содержит неполный набор признаков. На каждой выборке обучается одна модель, а при классификации каждая из них делает свой выбор в пользу одного из классов. Итоговое решение принимается с помощью усреднения набора решений всех моделей.

Бустинг – это способ объединения нескольких моделей, при котором каждая модель добавляет к исходным признакам своё решение, и следующая модель работает уже с дополненными исходными данными.

Стэкинг − это способ объединения нескольких моделей, при котором каждая модель принимает своё решение о классе объекта, а затем с помощью ещё одной модели, принимающей на вход уже ответы всех предыдущих моделей, выбирается итоговый класс объекта.

Есть методы AutoML, Рассказать про TPOT. <http://epistasislab.github.io/tpot/>

Преимущества:

* точнее, чем отдельная модель

Недостатки:

* Сложная архитектура требует больше времени на разработку
* Сложность интерпретации

<https://www.soa.org/globalassets/assets/files/e-business/pd/events/2020/predictive-analytics-4-0/pd-2020-09-pas-session-014.pdf>

1. **Применение МО (классификации) в ИБ**

В ИБ МО может применяться в случаях, когда сложно выявить детерминированные правила для работы системы – например, в случае, когда система сложна для понимания, или есть повышенные требования к скорости генерации таких пракил.

Выявление спама

Эвристические алгоритмы антивирусов

Контроль информационных потоков:

* DLP
* URL filtering
* NGFW

1. **Спам: определение, задача, существующие решения, их недостатки**

Спам (нежелательная корреспонденция) – телематическое электронное сообщение, предназначенное неопределенному кругу лиц, доставленное абоненту и (или) пользователю без их предварительного согласия и не позволяющее определить отправителя этого сообщения, в том числе ввиду указания в нем несуществующего или фальсифицированного адреса отправителя.

Страдает свободное или рабочее время человека, которое легко конвертировать в финансовые потери, а во втором есть риск понести прямые убытки, случайно передав злоумышленникам конфиденциальную информацию.

1. **Спам: решение с помощью МО**

В задаче классификации спама исходных данных достаточно – сэмплы генерируют злоумышленники, а размечают их занимаются сами пользователи своим поведением.

Прежде чем эти данные «скармливаются» классификатору, следует решить еще одну проблему — очистить их от шумов. Ведь основная сложность фильтрации спама заключается в том, что критерии восприятия полезности сообщений у разных людей могут быть разными. Один пользователь будет воспринимать письма с предложениями по распродажам как жесткий спам, в то время как другой сочтет их потенциально полезной информацией. Письма такого рода создают шумы, затрудняющие построение качественного алгоритма машинного обучения. Говоря языком статистики, в нашем наборе данных могут быть так называемые выбросы, то есть результаты, резко выделяющиеся из общей выборки. И для решения этой проблемы мы внедрили автоматическую фильтрацию выбросов на базе адаптированного под наши нужды алгоритма Isolation Forest. Безусловно, она позволяет отсеивать только часть шумов, но это уже значительно упрощает жизнь нашим алгоритмам.

В результате мы получаем практически «чистые» данные. Наша следующая задача — преобразовать их в формат, понятный классификатору. То есть в набор признаков (или, как мы говорим, фичей). По большому счету можно выделить три основных типа таких фичей, используемых в нашем классификаторе:

Текстовые фичи. С ними все более и менее понятно. Они представляют собой отрывки текста, которые часто встречаются в спам-письмах. После предварительной обработки этого текста (препроцессинга) мы получаем достаточно устойчивые признаки.

Экспертные фичи. Это признаки, базирующиеся на экспертных знаниях, накопленных нашими базами данных за долгие годы работы. Связанные, например, с доменами, с частотностью служебных заголовков и т.д.

Все эти признаки и их различные комбинации помогут нам на последнем шаге – запуске классификатора.

Напомню, на выходе мы хотим получить систему, которая допускает минимум ложных срабатываний, быстра и при этом выполняет основное свое предназначение – ловит спам. Для этого мы строим ансамбль классификаторов, и для каждого набора признаков он свой. В результате, вердикты всех классификаторов объединяют, и система выносит финальный вердикт.

<https://habr.com/ru/company/vk/blog/476714/>

<https://securelist.ru/machine-learning-versus-spam/29962/>

1. **Эвристические алгоритмы антивирусов: задача**

Эвристический анализ – это метод поиска вирусов, заключающийся в выявлении последовательностей команд, предположительно присущих вредоносной программе. Несмотря на возможные ошибочные решения, эвристический анализ позволяет существенно повысить эффективность работы антивирусного сканера как по обнаружению новых видов вирусов, так и полиморфных вирусов и вирус-генераторов, которых он может выявлять по характерным алгоритмам или вызовам внешних процедур.

1. **Эвристические алгоритмы антивирусов: МО**

Предположим, что рассматриваемый инструмент анализирует программу с помощью некоторых элементарных единиц, которые отражают поведение программы, например, машинные команды или системные вызовы. Машинная команда или системный вызов имеют смысловое значение. А это значит, что модель, которая состоит из статистически обработанных системных вызовов или машинных команд, тоже имеет некоторую интерпретацию. Аналогично тому, как смысл слова не равен смыслу каждой отдельной его буквы, так и понимание составных частей не дает понимания всей модели. Последовательность букв приобретает смысл лишь после того, как человек императивно назначит ему смысловую нагрузку.

<https://www.okbsapr.ru/library/links/obnaruzhenie-anomalnogo-povedeniya-programm-dlya-dalneyshego-ispolzovaniya-pri-reshenii-zadachi-zashch/>

<https://download.geo.drweb.com/pub/drweb/windows/workstation/12.0/documentation/html/av/ru/index.html?intro_detectionmethods.html>

1. **DLP: задача**

DLP-системе необходимо понимать, какие данные следует защищать. Как система определяет, что информация относится к конфиденциальной?

С самого начала развития DLP-системы опирались на характеристики файлов – время создания, метки и прочее. Можно по ключевым словам определить, содержит ли файл ценную информацию.

В последнее время всё активнее применяется «искусственный интеллект» в виде моделей машинного обучения, нейронных сетей. Они обучаются на известных документах разной степени ценности, затем сами предсказывают ценность обнаруживаемых документов, а при корректировке предсказаний сотрудниками отдела ИБ – дообучаются.

1. **DLP: применение МО**

Определение конфиденциальности новой информации – классификация на основе известных наборов данных.

Важные признаки для определения необходимости защиты файла – его свойств и контекст. Контекст – это дополнительные сведения о передаваемой информации: кто, когда, откуда, по какому каналу, как часто. Например, получение сотрудником, имеющим допуск к конфиденциальной информации, одного или двух документов в течение часа – событие рядовое, а вот запрос всей защищаемой библиотеки сразу – повод для сотрудника отдела ИБ проверить его.

1. **URLfiltering: задача**

Кроме контроля информации, выходящей из системы, надо контролировать ещё и входящую. В том числе – не допускать загрузку данных с URL-адресов, которые могут быть вредоносными

Существующие реализации – списки запрещённых адресов, списки частей URL-адресов, анализ сертификатов

Проблема – вредоносные URL-адреса становятся доступными раньше, чем появляются в списках запрещённых

<https://meliorit.ru/palo-alto-networks/subscription/url-filtering>

1. **URLfiltering: применнеие МО**

Известные признаки – элементы URL-адреса, HTML и JavaScript кода страницы

Вариант реализации – по списку запрещённых адресов определить зависимости между частями URL-адресов, соответствующие «зловредным» и «не-зловредным» Интернет-ресурсам

Для первого элемента исследуемого адреса предсказывать моделью, обученной на лексике «зловредных» ресурсов, следующую часть. Если предсказание подтвердилось – повысить вероятность того, что ресурс зловредный

Лексическую модель расширить с помощью известных данных из WHOIS и списков запрещённых ресурсов

В результате модель формирует итоговое решение.

<https://vixra.org/pdf/1906.0402v1.pdf>

1. **NGFW: задача**

Традиционные межсетевые экраны отстают от требований по обеспечению полной защиты

Универсальный шлюз безопасности (UTM) - устройство «всё в одном», объединившее возможности антивируса, VPN, межсетевого экрана, контент- и спам-фильтров, систем обнаружения и предотвращения вторжений

Проблема: UTM медленные и обнаруживают не все угрозы

Решение: создать FPGA-чипы и связать все сервисы в один для совместной обработки данных. Результат назвали NGFW – Next Generation FireWall

Но как связать все данные, ничего не пропустив?

Для решения этой задачи уместно использование ИИ

<https://www.smart-soft.ru/blog/chto_takoe_utm_ngfw_i_chem_oni_otlichajutsja/>

1. **NGFW: применение МО**

Новое устройство появилось в сети. Что это? Классификация по слепку сетевого трафика даст ответ

Много данных, непонятно, как составить правила для связи разных систем в составе NGFW? Подход «предобработка – выбор модели – оценка метрик» может помочь создать рекомендации

Непонятно, как защищаться от новых атак? Рассмотренные выше для антивирусов эвристические алгоритмы здесь также эффективны

<https://blog.tiger-optics.ru/2021/01/iot-security/>

<https://www.paloaltonetworks.com/cyberpedia/what-is-an-ml-powered-ngfw>

1. **Результат изучения модуля**

МО: задачи и направления

Базовые методы классификации и регрессии

Методы ансамблирования в МО

Применение в реальных задачах

1. **Спасибо за внимание**