Машинное обучение, ФКН ВШЭ
Семинар
Интерпретируемость моделей

1 Введение

Ранее в курсе мы рассматривали различные модели машинного обучения, основная задача которых сделать предсказание, «хорошее» с точки зрения некоторых функционалов качества. Чаще всего модели не могут сделать идеальных предсказаний (из-за шума в данных или отсутствия реальной функциональной зависимости между признаками и целевой переменной). В некоторых задачах, как например «ранжирование фильмов в рекомендациях», ошибка не так страшна и может вызвать лишь небольшой негатив у пользователя. Однако есть задачи, в которых цена — человеческие жизни или хотя бы большие суммы денег.

В случае, когда от модели зависит выбор способа лечения при диагностировании заболевания, становится важным понимать, почему модель приняла то или иное решение. По законам некоторых стран нельзя не давать кредит на основании модели, объяснить которую невозможно.

Также иногда в процессе разработки модели может быть полезно лучше понимать, на какие данные она «обращает внимание». Так можно найти ошибки в реализации подсчёта признаков (признаки не используются или используются не так, как ожидалось) или найти ошибки данных (например, утечки целевой переменной).

На этом семинаре рассмотрим несколько подходов к интерпретации моделей.

2 Интерпретация, основанная на особенностях моделей

Некоторые модели из-за своих особенностей позволяют понять, как они получают свои предсказания. Рассмотрим несколько из них.

1. Линейные модели. Естественной важностью признаков в линейных моделях являются веса (в случае нормализации признаков). По абсолютному значения можно судить о силе влияния на предсказание, а по знаку на направление. Однако в случае большого количества признаков или при наличии взаимосвязи между признаками могут быть искажения (например, два скоррелированных признака разделят между собой важность, а иногда один из них может иметь противоположный знак).

2. Деревья решений. Деревья легко визуализируются, однако в случае ансамбля (случайный лес или бустинг) визуализировать тысячи деревьев уже не представляется возможным.

3. Нейронные сети на изображениях. Для изображений можно строить карты уверенности предсказаний, закрывая часть изображения. Тогда можно будет увидеть части исходной картинки, из-за которых сеть делает свои предсказания.

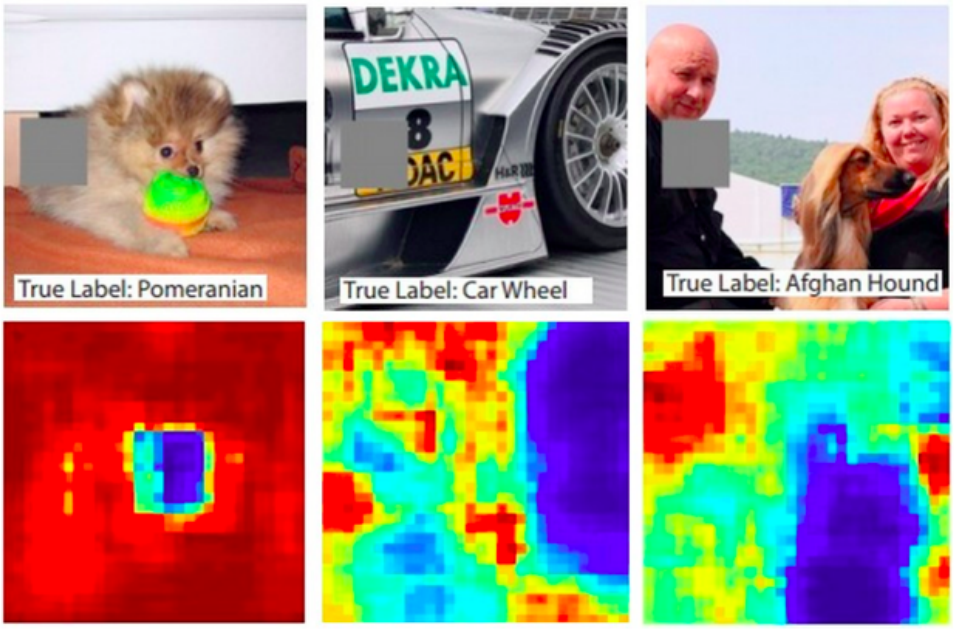

Рис. 1. Пример построения карт для нейронных сетей

3 LIME

Рассмотрим метод локальной интерпретации моделей через суррогатные модели (Local Interpretable Model-Agnostic Explanations). Основная идея такого подхода заключается в том, что мы аппроксимируем предсказания **объясняемой** модели в окрестности некоторого **объекта** с помощью простой **объясняющей** модели. Простая объясняющая модель достаточно простая, чтобы допускать интерпретацию, поэтому с помощью неё мы будем интерпретировать объясняемую модель.

Математически локальные суррогатную модель можно описать следующим образом:

$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g),$$

где x — объект, в окрестности которого интерпретируется модель, L — функция потерь для аппроксимации (например, квадратичная), G — семейство возможных объясняющих моделей, $\Omega(g)$ — сложность объясняющей модели, π_x — мера окрестности объекта x для аппроксимации.

В LIME выбор сложности объясняющей модели выбирается пользователем (количество используемых признаков), в качестве семейства моделей берётся Lasso-регрессия (из-за ограничения на число признаков), для которой подбирается коэффициент регуляризации для достижения необходимого числа признаков, или дерево решений.

Построение суррогатной модели происходит по следующей схеме:

1. Выбрать объект x, в окрестности которого производится аппроксимация.
2. Сгенерировать новые объекты в окрестности объекта x.
3. Обучить объясняющую модель на сгенерированных точках (с учётом ограничения на сложность).
4. Интерпретировать обученную объясняющую модель.

Способ получения новых сгенерированных объектов зависит от их типа:

1. Для текстовых данных в случае Bag-of-words представления можно исключать отдельные слова из объекта x (аналогично бинарные данные).
2. Для изображений можно отключать часть пикселей из объекта x.
3. Для табличных данных с вещественными признаки генерируются новые объекты, в которых признаки искажаются в соответствии с нормальным распределением вокруг объекта.

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Рис. 2. Пример работы LIME для описания 3 классов

Каждому сгенерированному объекту могут приписываться веса, с которыми они входят в функцию потерь L. Эти веса обычно рассчитываются с помощью ядер как расстояние от сгенерированного объекта до объекта x.

У LIME существуют проблемы с данными большой размерности или с признаками коррелирующими друг с другом. Для табличных данных предполагается отсутствие взаимосвязей между признаками, что далеко не всегда верно.

4 Influential Instances (влиятельность объектов)

Альтернативой подходу с оценкой влияния признаков и их изменения на предсказания объясняемого алгоритма является подход с оценкой влияния объектов обучающей выборки на построенный алгоритм. Существует два подхода для оценки влияния объектов: через удаление объекта из обучающей выборки и через оценку влияния на функцию потерь.

Оценка влияния объектов на модель помогает ответить на вопросы об устойчивости моделей (хорошая модель не должна сильно зависеть от отдельных объектов) и найти вопросы в данных, которые могут мешать настройки модели (модели, плохо устойчивые к выбросам, сильно изменяются при их удалении).

§4.1 Диагностика через удаление

Для оценки влиятельности объекта обучим модель 2 раза: на полной обучающей выборки и на обучающей выборке без этого объекта. Теперь можно оценить, насколько обученные модели отличаются друг от друга. Если у модели есть явный вектор параметров, то можно посчитать отличия в векторах параметров (например, как норму разности). Более универсальный подход заключается в сравнении предсказаний двух моделей, например, с помощью расстояния Кука (Cook's distance). Для задачи регрессии влияние объекта i можно оценить как:

$$D_i = \frac{\sum_{j=1}^{l}(\hat{y}_j - \hat{y}_j{}^{-i})^2}{d \times \text{MSE}},$$

$\hat{y}_j$ — предсказание модели, обученной на полных данных на объекте j, $\hat{y}_j^{-i}$ — предсказание модели, обученной без объекта i на объекте j, d — количество признаков, MSE — исходная квадратичная ошибка модели.

На практике поиск влиятельных объектов может быть мало информативным — сложно интерпретировать таблицу с признаками десяти самых влиятельных объектов. Однако можно построить простую модель (например, решающее дерево), которые будет детектировать эти влиятельные объекты относительно всех остальных объектов. Так можно будет увидеть, что особенного в этих влиятельных объектах есть.

Также можно оценивать влиятельность конкретного объекта на предсказание другого объекта, оценивая разность от предсказаний двух моделей.

§4.2 Функции влияния (influence functions)

Недостатком подхода с удалением является его сложность — на практике обучить моделей столько же, сколько объектов в обучающей выборке не представляется возможным. Альтернативный вариант предлагает для моделей с дифференцируемой функцией потерь исследовать влияние изменения веса для конкретного примера обучающей выборки на параметры модели через влияние на функцию потерь.

Обозначим за $\hat{\theta}_{\epsilon,z}$ параметры модели, если вес у объекта z увеличить на ϵ:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta}(1-\epsilon)\frac{1}{l}\sum_{i=1}^{l} L(z_i,\theta) + \epsilon L(z,\theta)$$

Тогда влияние объекта z на ошибку на объекте z_{test} можно следующим образом:

$$\begin{aligned} I_{\text{up,loss}}(z, z_{\text{test}}) &= \frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}|_{\epsilon=0} \\ &= \nabla_\theta L(z_{\text{test}}, \hat{\theta})^T \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^T H_\theta^{-1} \nabla_\theta L(z, \hat{\theta}), \end{aligned}$$

где гессиан можно посчитать приближённо.

Интуиция за этой формулой следующая. Представим гессиан равным единичной матрице, тогда положительная влиятельность объекта z на функцию потерь (ухудшение) — это противоположные направления градиентов функции потерь для объектов z и z_{test} (то есть объекта z «мешает»).

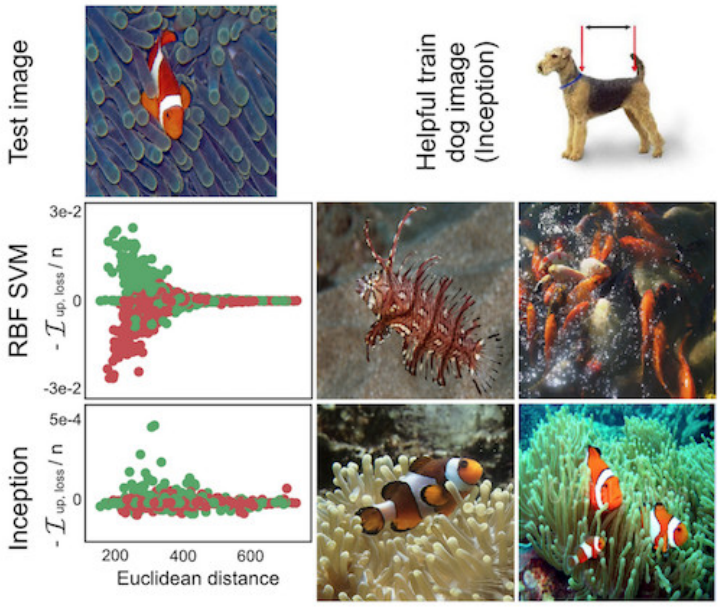

Рис. 3. Визуализация поиска влиятельных изображений для задачи классификации картинок на рыбы и собаки. Для метода опорных векторов влиятельные изображения это рыбы похожих цветов, а нейронная сеть обращает больше внимания на форму. Видно, что нейронная сеть ищет более сложные паттерны (большее евклидово расстояние между изображениями)

Использовать функции влияния можно несколькими способами:

1. Сравнивать модели между собой. Как на примере с изображением может получиться так, что одна из моделей ищет более сложные паттерны на изображении по сравнению с другой. Для этого нужно сравнивать между собой наиболее влиятельные изображения для некоторого примера выборки.

2. Детектирование несовпадений доменов между обучающим и тестовым множеством. Можно найти ложное срабатывание модели и изучить влиятельные объекты для этого ложного предсказания. Так можно выяснить паттерны в данных, которые мешают корректной работе алгоритма на новом домене данных (данные из несколько другого распределения).

3. Коррекция обучающих данных. Если у нас есть возможность перепроверить корректность разметки небольшого числа объектов обучающей выборки, то эффективнее сделать это на наиболее влиятельных объектах, так как именно они влияют на нашу модель сильнее всего.

5 Adversarial примеры

Кроме интерпретации модели для оценки её устойчивости через признаки или обучающие примеры, можно оценивать корректность модели через её устойчивость к adversarial примерам. Adversarial примерами называют такие примеры, которые при малом отличии от исходных примеров заставляют алгоритм делать другие предсказания. Например, при наложении шума на картинку с кошкой можно получить предсказание собаки.

Возникает вопрос, как генерировать такие примеры. Существует два основных подхода к генерации таких примеров: с доступом к градиенту модели и black box, то есть без доступа к градиенту модели.

Например, можно генерировать картинку, решая следующую оптимизационную задачу:

$$L(f(x+r), l) + c\|r\|,$$

где L — функция потерь нашей модели, f — модель, x — атакуемое изображение, r — добавка, которую мы ищем, l — желаемый класс, c — коэффициент, балансирующий штраф за размер надбавки.

В случае с black box атаками учатся сурогатные модели на предсказаниях чёрного ящика, для которого далее производится атака методами, использующими градиент.