МАШИННОЕ ОБУЧЕНИЕ

Катя Герасименко, Саша Ершова

January 11, 2019

ЗОШ-2019

Качество модели

От чего зависит качество модели?

- Данные
- Признаки
- Сама модель
- Гиперпараметры
- Рандом

ДАННЫЕ

Иногда данные бывают такие, что f-score 0.4 — это лучшее, чего можно добиться.

- Шумные данные
- Мало данных
- Очень размытая и непонятная целевая переменная

Фичи

Признаки — features — фичи
Половина (или больше) успеха
Фичи — очень важно, нужно внимательно на них смотреть и думать, что из них можно получить.

FEATURE ENGINEERING

Создаем фичи:

- из нескольких имеющихся фичей — новая. Пример — что-нибудь сложить
- переклассифицировать фичи. Пример — из конкретных «рядом есть кафе / ресторан / закусочная / банк / магазин / аптека / больница / школа» сделать признак «рядом есть еда»
- данные из внешних источников. Пример — рейтинг ресторана, найденный в интернете

Feature engineering для текста

Что можно вытащить из текста?

FEATURE ENGINEERING ДЛЯ ТЕКСТА

Что можно вытащить из текста?

- слова (CountVectorizer, TfidfVectorizer)
- n-граммы
- символьные n-граммы
- длину текста
- наличие специфических символов и их класс (например, смайлики - веселые, грустные)
- морфологическая информация: части речи, одушевленность, countability...
- синтаксическая информация: какие есть связи между группами, какая вершина у вышестоящей группы
- семантическая информация: сем.роли, гиперонимы: WordNet
- вектора из дистрибутивных моделей
- все вышеперечисленное для левого и правого контекста

РАЗМЕРНОСТЬ

Нагенерили миллион признаков, можно радоваться? Нет.
Много признаков, особенно когда мало данных — зло.

- Модель долго обучается
- Много шума, что зашумляет модель и повышает ошибку
- Модель может переобучиться

С этим можно бороться.

- Feature selection
- Feature extraction

FEATURE SELECTION

Давайте выберем самые важные признаки.

- признаки, наиболее соотнесенные с целевой переменной (корреляция, mututal information)
- признаки, лучше всего предсказывающие целевую переменную (обучить на подмножестве признаков, посмотреть качество)
- использовать модели, которые дают веса признаков или при обучении обнуляют неважные признаки (L1-регуляризация)

FEATURE EXTRACTION

Давайте из кучи старых признаков сделаем не такую кучу новых.

- PCA
- SVD

САМА МОДЕЛЬ

Разные модели работают хорошо на разных данных.
Пример: дерево решений долго и не очень хорошо справляется с разреженными данными с кучей признаков.
Почему?

ГИПЕРПАРАМЕТРЫ

У модели есть параметры, которые настраиваются внутри нее.
А есть гиперпараметры — параметры, которые выставляются датасатанистом.
Примеры:

- количество эпох (проходов по всем данным) в нейросети
- размер батча (кусочка данных, который подается на вход) в нейросети
- количество деревьев в случайном лесу
- тип регуляризации (L1, L2)

и много других параметров, которые зависят от конкретных моделей

ЕГО ВЕЛИЧЕСТВО РАНДОМ

Многие модели инициализируют свои параметры / веса случайно.
Это значит, что каждый запуск обучения завершится с немного разными значениями параметров. Что с этим делать:

- как минимум — фиксировать. Если видите random_state или его аналоги — сразу ставьте.
- хорошая идея: обучить модель с несколькими значениями random_state и усреднить предсказания, чтобы убрать влияние конкретного random_state на наши результаты.

Оценка качества модели

КАК ПРОВЕРЯТЬ МОДЕЛЬ?

Валидация на обучающей выборке

Не надо так делать.

Максимум — чтобы проверить, что модель просто обучилась, и в данных не наблюдается полного отсутствия закономерностей.

TRAIN/TEST

1. Разбиваем выборку на две неравных части (например, 2/3 и 1/3)
2. Меньшую часть откладываем и не трогаем
3. Обучаем модель на большей части (train set)
4. Предсказываем значения на тестовой части (test set)
5. Сравниваем получившиеся значения с реальными, при необходимости тюним модель

TRAIN / VALIDATION / TEST

1. Разбиваем выборку на три неравных части (одна большая, две маленьких)
2. Обучаем модель на train set
3. Предсказываем значения на validation set, при необходимости тюним параметры
4. Смотрим итоговое качество модели на test set

КРОССВАЛИДАЦИЯ

k-folds

1. Разбиваем выборку на k частей
2. Используем одну часть как тестовую выборку, остальные — как обучающую
3. Повторяем k раз — по одному для каждой части
4. Усредняем полученную оценку

Feature extraction для текста

ПОДБОР ГИПЕРПАРАМЕТРОВ

GRIDSEARCH

1. Выбираем, какие гиперпараметры модели хотим тюнить.
2. Для каждого гиперпараметра задаём список проверяемых значений.
3. Обучаем модели со всеми возможными комбинациями гиперпараметров, проверяем каждую кроссвалидацией, выбираем лучшую.

Это работает **очень долго**. Если, например, у вас есть три параметра, и для каждого параметра хочется проверить 5 значений, всего придётся обучить 125 моделей.

RANDOMIZEDSEARCH

Обучает N случайных комбинаций гиперпараметров, а не все, как GridSearch. Может не найти самое оптимальное значение, но работает быстрее.

Ансамблевые методы

Одна модель — хорошо, а много — лучше.
Как можно объединить модели?

ТИПЫ АНСАМБЛЕЙ

- голосование
- стэкинг
- бустинг

ГОЛОСОВАНИЕ

Возьмем несколько моделей, предскажем ими результат по нашим данным, совместим результаты (среднее, среднее с весами, наиболее частый ответ)

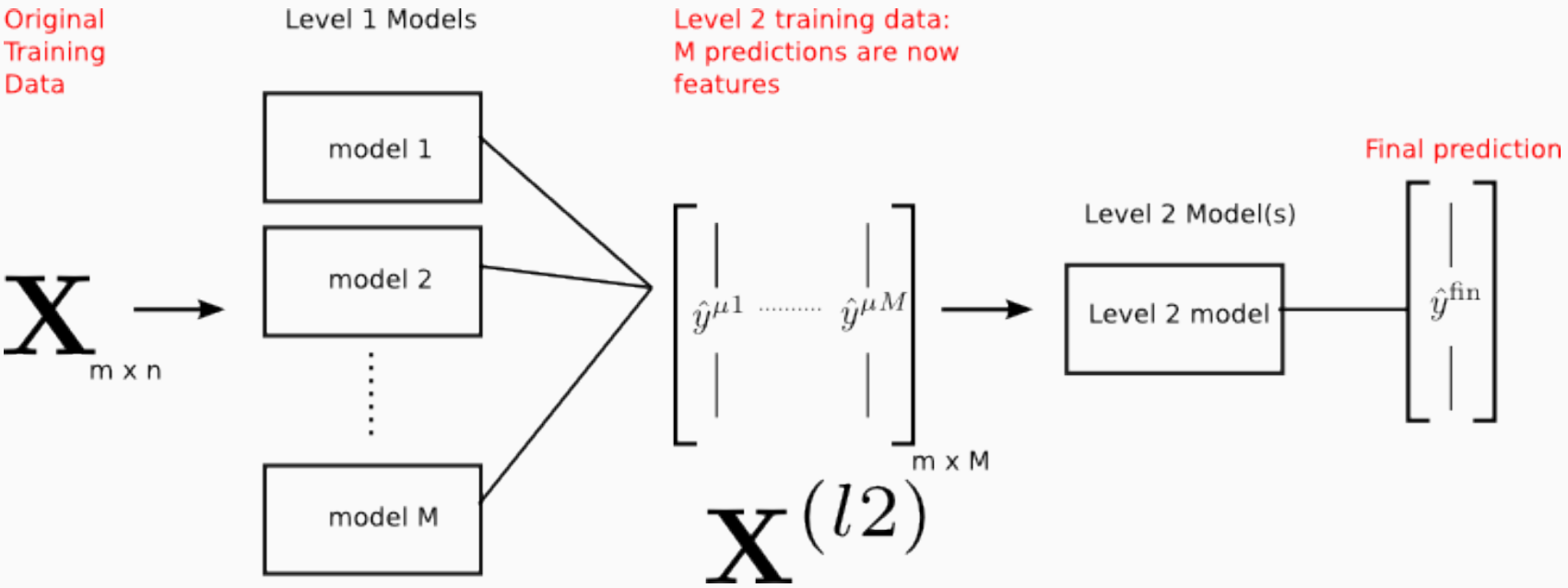
bootstrap aggregation – bagging: можно обучать модели

- На части объектов
- На части признаков

Привер: Random Forest. Одно дерево склонно к переобучению. Давайте сделаем голосование деревьев.

СТЭКИНГ

Возьмем несколько моделей, предскажем ими результат, предсказания используем в качестве признаков в другой модели.



БУСТИНГ

Возьмем наши данные и начнем обучать на них слабые модели (e.g. решающие пни).
После обучения слабой модели учитываем качество ее предсказаний.
Перенастраиваем веса объектов — неправильно предсказанным даем больший вес, правильно предсказанным — меньший.
С новыми весами обучаем следующую слабую модель.
...
PROFIT!

Спасибо за внимание!