Попытка сравнения библиотеки CatBoost с другими открытыми библиотеками градиентного бустинга

1 Немного про CatBoost

CatBoost – open source библиотека градиентного бустинга, которую два месяца назад презентовал Яндекс. Если верить разработчикам, она способна решить все наши проблемы: Кэтбуст устойчив к переобучению, может работать с категориальными признаками без дополнительной предобработки и работает лучше других аналогичных открытых библиотек. Вместе с исходным кодом Яндекс выложил серию экспериментов, в которых на наборе из 9 публичных датасетов CatBoost бьет и всем известный XGBoost, и майкрософтовский LightGBM, и библиотеку H2O - причем как после подбора гиперпараметров, так и с дефолтными параметрами (по оптимизируемой метрике LogLoss).

Рис. 1: Эта картиночка здесь, потому что Никита любит дурацкие картиночки...

Итак, что умеет эта библиотека. Кэтбуст поддерживает несколько вариантов преобразования категориальных фичей в чиселки, и это не one-hot encoding, а подсчеты статистик. One-hot encoding тоже поддерживается, если очень уж хочется, но это нужно указывать в стартовых параметрах (может быть полезно, если категорий не очень много, так как подсчет статистик требует больше времени и памяти).

Создатели пишут, что алгоритм устойчив к подбору параметров, так как он тестировался на большом количестве разных наборов данных и, соответственно, разрабатывался так, чтобы на разных данных было хорошее качество. Поэтому предполагается, что его можно использовать, не тратя время и силы на перебор гиперпараметров, а в дефолтной конфигурации - и сразу будет хорошо.

Библиотека написана на C++ и работает из-под Питончика и R. Кэтбуст строит oblivious деревья решений, а для борьбы с переобучением Яндекс придумал свою реализацию алгоритма градиентного бустинга, которая некоторым хитрым образом уменьшает смещения остатков.

Еще Кэтбусту можно указать пользовательскую целевую функцию. К нему также прилагается тул для визуализации процесса обучения, который можно запустить отдельно в браузере или в тетрадке Jupyter.

Но есть один большой минус – это скорость работы. По моим предварительным наблюдениям, сразу после выхода Кэтбуст отставал от своих аналогов по этому параметру в десятки раз.

Рис. 2: Эта картинка иллюстрирует, что не я одна это заметила.

Авторы сразу говорили, что сравнивать время работы рано и что работа над ускорением в процессе. И вот пару недель назад Яндекс выложил новую версию своей библиотеки, которая должна по крайней мере частично решить эту проблему.

Кроме скорости, были доработаны и другие параметры. Например, теперь Кэтбуст просто, но элегантно поддерживает работу с пропущенными значениями, а тул для визуализации интегрируется в TensorBoard.

2 Расследование

Мы решили проверить, собственно, так ли хорош CatBoost на самом деле и сравнили его на трех совершенно случайно выбранных датасетах с библиотеками XGBoost и LightGBM.

Я следовала той же идеологии, что и Яндекс в своих бенчмарках, и использовала их же код, так что общая идея - сравнить работу алгоритмов "из коробки"и проверить заявление о том, что Кэтбуст хорош и без подбора гиперпараметров: я ничего не оптимизировала, а просто просто потестировала все алгоритмы с их дефолтными параметрами, тюня только количество деревьев.

При этом подбор деревьев проходил в два этапа: с ограничением на максимальное число в 100 деревьев (чтобы посмотреть, какой алгоритм справляется лучше совсем небольшим количеством деревьев (на самом деле потому что мне было лень ждать)) и в 5000 деревьев, как делал Яндекс (чтобы дать Кэтбусту второй шанс, потому что ОСТОРОЖНО СПОЙЛЕР в предыдущем этапе победил не он).

Оптимизировала тоже LogLoss и еще зачем-то смотрела на F-меру по классам. И фиксировала, сколько времени каждому алгоритму требуется на обучение итоговой модели (среднее значение по пяти итерациям, на моем компуктере i5-6600K, 4 CPUs, 32 GB RAM).

Ноутбуки лежат тут.

2.1 Версии библиотек

♡ CatBoost: 0.2

♡ XGBoost: 0.6

♡ LightGBM: 2.0.5

3 Данные

3.1 Credit Card Fraud

Предсказание кражи денег с кредитных карт [1]

В этом датасете 30 числовых признаков, 28 из которых – результат применения метода главных компонент к первоначальным признакам, информация о которых не доступна по причинам конфиденциальности. Это задача бинарной классификации: зависимая переменная принимает два значения: *все ок* (0) vs. *произошла кража* (1). Датасет очень несбалансированный: в нем всего 284807 строк (транзакций), из которых только 492 классифицированы как кража (меньше 0.2% всех транзакций).

3.2 IMDB reviews

Анализ тональности рецензий с сайта IMDB

Это наполовину игрушечный датасет с текстовыми данными – ревью фильмов с сайта IMDB. Наполовину, потому что оценка сведена к бинарной: рейтинг < 5 считается отрицательной оценкой (0), рейтинг $>= 7$ – положительной (1). А еще потому что количество объектов обоих классов одинаковое (всего 25000 наблюдений).

Я не делала никакой дополнительной предобработки текста (кроме дефолтных удаления стоп-слов и приведения к нижнему регистру), а для преобразования текста в набор признаков использовала HashingVectorizer со значением количества признаков 5000 (такое число кажется достаточным, так как, например, логистическая регрессия дает F-меру 0.85 на тестовой выборке (20% от датасета)).

3.3 Shelter Animals

Предсказание судьбы питомца в приюте

Этот датасет содержит данные о животных (кошках и собаках) в приютах Америки с 2013 по 2016 года. В нем есть категориальные фичи: тип животного, пол, порода, цвет; и еще информация о дате и времени сдачи питомца в приют и о его возрасте. Я делаю минимальную предобработку даты и возраста для преобразования их в числовые признаки. Целевая переменная принимает пять значений: питомец может умереть, быть подвергнут эвтаназии, его могут передать в другой приют, или найти новые хозяева, или забрать обратно старые.

На самом деле это задача многоклассовой классификации. Но попытка запустить Кэтбуст на такой таск провалилась (он ушел в отрицательный логлосс – баг?), поэтому исходы были сведены к бинарным: все хорошие исходы (*передача, возврат, новые хозяева*) – 0, плохие (*смерть, эвтаназия*) – 1.

И это тоже несбалансированный датасет, из 26729 наблюдений меньше 7% принадлежат к "плохому"классу.

4 Результаты

4.1 Credit Card Fraud

В минимальной версии Кэтбуст справляется чуть хуже и по LogLoss, и по F-мере для важного миноритарного класса. С большим числом деревьев Кэтбуст сразу исправляется, но обучает модель в 16 раз медленнее, чем XGBoost, и в 28 раз медленнее, чем LightGBM.

[1] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In *Symposium on Computational Intelligence and Data Mining (CIDM)*, IEEE, 2015

	LogLoss	F-measure (Класс 0)	F-measure (Класс 1)	Среднее время работы (с)	Число деревьев
n_estimators = 100					
Default CatBoost	0.00245	0.99977	0.85455	27.85	100
Default XGBoost	0.00237	0.99977	0.85870	6.34	32
Default LightGBM	0.00238	0.99977	0.85714	1.95	100
n_estimators = 5000					
Default CatBoost	0.00218	0.99979	0.86792	88.28	723
Default XGBoost	0.00237	0.99977	0.85870	5.56	32
Default LightGBM	0.00236	0.99977	0.85870	3.10	202

Таблица 1: Результаты работы библиотек на датасете Credit Cards Fraud

4.2 IMDB reviews

В первом этапе Кэтбуст проигрывает довольно значимо - композиции 100 деревьев на 5000 фичей и 25000 строк ему явно сильно не хватает. Во втором этапе он значительно улучшается и всех обгоняет (увеличив число деревьев в 50 раз), но работает при этом настолько долго, что вообще непонятно, стоит ли улучшение в 0.01 метрики таких страданий (например, на обучение модели он тратит в 30 раз больше времени, чем XGBoost, и в 334 (!!!) – чем LightGBM).

	LogLoss	F-measure (Класс 0)	F-measure (Класс 1)	Среднее время работы (с)	Число деревьев
n_estimators = 100					
Default CatBoost	0.50903	0.75104	0.79194	135.00	100
Default XGBoost	0.37462	0.83035	0.83940	65.03	100
Default LightGBM	0.34905	0.84435	0.84683	16.48	100
n_estimators = 5000					
Default CatBoost	0.33280	0.85686	0.86123	6757.49	5000
Default XGBoost	0.35183	0.84507	0.85121	227.67	348
Default LightGBM	0.34620	0.84693	0.84984	20.26	130

Таблица 2: Результаты работы библиотек на датасете IMDB reviews

4.3 Shelter Animals

И с этим датасетом расклад примерно тот же: небольшим числом деревьев Кэтбуст справляется чуть хуже других библиотек, причем совсем плохо с определением миноритарного класса. На втором этапе Кэтбуст немного выходит вперед по оптимизируемой метрике, снова уступая в скорости работы в десятки раз.

	LogLoss	F-measure (Класс 0)	F-measure (Класс 1)	Среднее время работы (с)	Число деревьев
n_estimators = 100					
Default CatBoost	0.19716	0.96652	0.03232	1.34	100
Default XGBoost	0.18976	0.96787	0.21801	0.10	24
Default LightGBM	0.19312	0.96617	0.20230	0.13	53
n_estimators = 5000					
Default CatBoost	0.18753	0.96709	0.20881	15.21	1101
Default XGBoost	0.18976	0.96787	0.21801	0.18	24
Default LightGBM	0.19312	0.96617	0.20230	0.17	53

Таблица 3: Результаты работы библиотек на датасете Shelter Animals

5 Рандомные наблюдения

♡ Для преобразования категориальных фичей в числовые Кэтбуст использует подсчет счетчиков, и это хорошо.

♡ Кэтбуст пока что работает только с деревьями решений, нет возможности указать тип элементарных алгоритмов (у XGBoost, например, это реализовано).

♡ Мне так и не удалось запустить Кэтбуст на многоклассовую классификацию с подбором деревьев. Надеюсь, проблема во мне.

♡ Кэтбуст все еще очень долгий. ОЧЕНЬ. В среднем в 7-8 раз дольше, чем LightGBM, в 2-3 раза дольше чем XGBoost.

♡ Кэтбуст заявлен как алгоритм, умеющий работать с категориальными признаками без предобработки, при этом в задачах классификации Кэтбуст не умеет интерпретировать целевую переменную, если она задана строкой, а не числом. Вот такие пирожочки.

♡ Наблюдение, немного выходящее из ряда: два параметра, описанных как дефолтные в документации, не соответствуют тем, с которыми по умолчанию запускается алгоритм))

♡ Просто сердечко.

6 Выводы

Итак, Кэтбуст действительно привлекает своими возможностями и качеством работы. Соглашусь с безымянными экспертами Яндекса: выложить в опен-сорс такой проект – это маленький шаг для компании, но большой – для всего data science сообщества.

> "Yandex has a long history in machine learning. We have the best experts in the field. By open-sourcing CatBoost, we are hoping that our contribution into machine learning will be appreciated by the expert community, who will help us to advance its further development," says Name Surname, position at Yandex.

Однако кажется, что CatBoost справляется лучше других алгоритмов градиентного бустинга только с большим количеством деревьев (четырехзначного порядка). Конечно, сравнивать этот параметр вообще некорректно – Кэтбуст использует другие деревья, ему их нужно больше. Но это, видимо, сильно влияет на скорость работы, а работает Кэтбуст все еще сильно дольше аналогов. Вывод: использовать его может быть полезно в соревнованиях на Каггле и подобных, когда на счету каждая тысячная доля метрики, или если в распоряжении есть куча мощностей, но, возможно, на практике пока что лучше использовать другие библиотеки (например, LightGBM). Так что надеемся, что ребята еще поработают над ускорением своего творения, и ждем обновлений.