Московский Государственный Университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Курсовая работа

Кодирование категориальных признаков

Работу выполнил
студент 317 группы
Махин Артем Александрович

Научный руководитель
д.ф.-м.н., профессор
Дьяконов Александр Геннадьевич

Москва
2021

Содержание

1 Введение

Большинство алгоритмов машинного обучения позволяют работать лишь с вещественными признаками, тогда как в реальном мире часто в данных присутствуют категориальные признаки, которые принимают свои значения из некоторого конечного множества. В настоящей работе проанализированы некоторые методы кодировки категориальных признаков, а так же их сравнение на конкретных задачах бинарной классификации с использованием двух алгоритмов машинного обучения: градиентный бустинг и логистическая регрессия.

2 Постановка задачи

2.1 Основные понятия

2.1.1 Задача бинарной классификации

Опишем постановку задачи бинарной классификации. Пусть задано множество объектов $\mathcal{X}$, а так же множество меток $\mathcal{Y} \in \{0, 1\}$. Существует неизвестная целевая зависимость - отображение $y^* : \mathcal{X} \to \mathcal{Y}$, значения которой известны только на объектах конечной обучающей выборки $\{(X_1, y_1), ..., (X_n, y_n)\}$, $y_i = y^*(X_i)$, $X_i \in \mathcal{X}$. Пары (X_i, y_i) называются прецендентами, совокупность пар $(X_i, y_i)_{i=1}^n$ называется обучающей выборкой. Требуется, основываясь на обучающей выборке, построить алгоритм $a : \mathcal{X} \to \mathcal{Y}$, способный классифицировать произвольный объект $X \in \mathcal{X}$.

2.1.2 Оценка качества алгоритмов

Для оценки качества работы бинарного классификатора существуют различные функционалы. В данной работе для оценки качества будет использована метрика ROC-AUC. Пусть есть n объектов, истинный вектор меток у которых y. Пусть так же имеется вектор $\widetilde{y}$ предсказаний вероятностей принадлежности к классу '1'. Тогда

$$AUC(y, \widetilde{y}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} I[y_i < y_j] * I'[\widetilde{y_i} < \widetilde{y_j}]}{\sum_{i=1}^{n} \sum_{j=1}^{n} I[y_i < y_j]}$$

где

$$\text{I'}[\widetilde{y_i} < \widetilde{y_j}] = \begin{cases} 0 & \widetilde{y_i} > \widetilde{y_j} \\ 0.5 & \widetilde{y_i} = \widetilde{y_j} \\ 1 & \widetilde{y_i} < \widetilde{y_j} \end{cases}$$

$$\text{I}[\text{y}_i < y_j] = \begin{cases} 0 & y_i \geq y_j \\ 1 & y_i < y_j \end{cases}$$

AUC принимает свои значения от 0 до 1, имеет смысл вероятности верно отранжировать (у объекта с классом '1' предсказанная вероятность выше) два объекта разных классов, случайно взятых из выборки. Чем ближе значение к 1, тем выше качество алгоритма.

2.2 Постановка задачи с категориальными признаками

Во многих алгоритмах машинного обучения предполагается, что все признаки принадлежат вещественным числам ($X_i^m \in \mathbb{R}$), однако во многих задачах данные могут принимать свои значения из множеств, которые не являются подмножествами вещественных чисел. К примеру, марка автомобиля. Этот признак может принимать свои значения из множества {Lada, Kia, BMW, ...}. Такие признаки называются категориальными.

3 Используемые методы кодирования

3.1 Label encoder

Самым простым способом способом кодирования категориальных признаков является Label encoder. Задается биективное отображение между уникальными значениями признака (пусть их будет n) и целыми числами от 0 до n-1. При таком подходе могут возникнуть проблемы: изначальное описание объектов не упорядочено, тогда как итоговое описание упорядочено, при чем случайным образом. Вероятно, алгоритм будет учитывать этот порядок, что скажется не лучшим образом на результат.

3.2 Frequency encoder

В некоторых случаях можно предположить, что частота встречаемости значения категориального признака как-то связаны с целевой переменной. Тогда каждый категориальный признак можно закодировать частотой своей встречаемости в обучающей выборке.

3.3 Target encoder

Значения категориальных признаков так же можно кодировать используя информацию с целевой переменной. Среднее арифметическое целевых переменных переменных является достаточно хорошим значением для кодирования, однако количество элементов какого-то класса может оказаться небольшим, что, скорее всего, приведет к смещенному показателю среднего значения. В этом случае применяют регуляризатор: i-ое значение кодируется как

$$global\_mean * (1 - \alpha_i) + mean_i * \alpha_i$$

где $global\_mean$ - среднее целевой переменной по всем наблюдениям, $mean_i$ - среднее целевой переменной по всем наблюдением с i-ым категориальным

признаком. Коэффициент α можно выбирать по-разному. Пусть количество наблюдений с i-ым категориальным признаком равно N_i. В данной работе выбирается следующим образом:

$$\alpha_i = \frac{1}{1 + e^{-\frac{N_i}{smoothing}}}$$

где $smoothing > 0$ - выбираемый гиперпараметр (бОльшие значения ведут к более сильной регуляризации)

3.4 James Stain encoder

Данный метод схож с Target encoder, различие заключается в выборе коэффициента α_i. Пусть в обучающей выборке у целевой переменной выборочная дисперсия равна std, а у объектов с i-ым категориальным признаком выборочная дисперсия целевой переменной равна std_i. Тогда

$$\alpha_i = 1 - \frac{std_i}{std_i + std}$$

3.5 Weight of Evidence encoder

Предположим, что в обучающей выборке с конкретным категориальным признаком есть N_+ пар с истинной меткой '1' и N_- с меткой'0'. В данном методе категориальный признак кодируется как

$$ln(\frac{N_+}{N_-})$$

Однако, чтобы в случае отсутствия элементов с меткой '0' не появлялось деление на 0, в числитель и знаменатель добавляют небольшое число. Получается следующая запись:

$$ln(\frac{N_+ + 0.5}{N_- + 0.5})$$

4 Используемые модели

4.1 Gradient Boosting

В качестве первой модели выступает градиентный бустинг, реализация которого взята с библиотеки sklearn. Модель обучалась при различных количествах деревьев (перебор осуществлялся по логарифмической сетке) , максимальная глубина деревьев равна 4. Для оценки качества набор данных разбивался на две части, на первой модель обучалась, на второй тестировалась.

4.2 Logistic regression

В качестве более простой модели выбрана логистическая регрессия, реализация взята из библиотеки sklearn. Для оценки качества использовалась k-fold кросс-валидация с 5 фолдами: обучающая выборка делится на 5 частей, затем проводится 5 итераций; на каждой итерации модель обучается на 4 частях и тестируется на части, которая не входила в обучение, результаты со всех 5 частей усредняются.

5 Данные

Для экспериментов были выбраны четыре набора реальных данных с категориальными признаками. В данном разделе приведен краткий обзор этих наборов.

5.1 Click-Through Rate Prediction

Данный набор был опубликован на международном соревновании по анализу данных на платформе Kaggle в 2014 году. Далее в настоящей работе для краткости этот набор данных будет называться Click.

В данном наборе данных необходимо решить задачу бинарной классификации: предсказать, кликнет ли пользователь по данному баннеру или нет. Всего использовалось 150 тысяч объектов, из них классу '1' принадлежат $\approx 17.4\%$ объектов. На 120 тысячах производилось обучение, на 30 тысячах тестирование. Каждый объект описывается 22 признаками, все категориальные.

5.2 Amazon.com - Employee Access Challenge

Второй набор был опубликован на международном соревновании по анализу данных на платформе Kaggle в 2013 году. Далее в настоящей работе для краткости этот набор данных будет называться Amazon.

В качестве задачи необходимо предсказать потребность сотрудника компании в доступе в зависимости от должности. Всего использовалось 32.7 тысяч объектов, из них классу '1' принадлежат $\approx 94.2\%$ объектов. На 26.2 тысячах производилось обучение, на 6.5 тысячах тестирование. Каждый объект описывается 9 признаками, все категориальные.

5.3 OneTwoTrip Contest

Третий набор был опубликован на международном соревновании по анализу данных на платформе Boosters в 2019 году. Далее в настоящей работе для краткости этот набор данных будет называться OneTwoTrip.

В данном наборе данных необходимо решить задачу бинарной классификации: предсказать подачу заявки на возврат билета пользователем. Всего использовалось 196 тысяч объектов, из них классу '1' принадлежат $\approx 2.2\%$

объектов. На 156.8 тысячах производилось обучение, на 39.2 тысячах тестирование. Каждый объект описывается 40 признаками, 29 из которых категориальные.

5.4 Porto Seguro's Safe Driver Prediction

Четвертый набор был опубликован на международном соревновании по анализу данных на платформе Kaggle в 2017 году. Далее в настоящей работе для краткости этот набор данных будет называться Driver.

Данный набор данных содержит большое число категориальных признаков (50 категориальных из 57). Необходимо предсказать то, что водитель подаст иск об автостраховании в следующем году. Весь набор содержит 595 тысяч объектов, 476 тысячи из которых были для обучения, 119 тысяч для тестирования. Классу '1' принадлежат $\approx 3.6\%$ объектов.

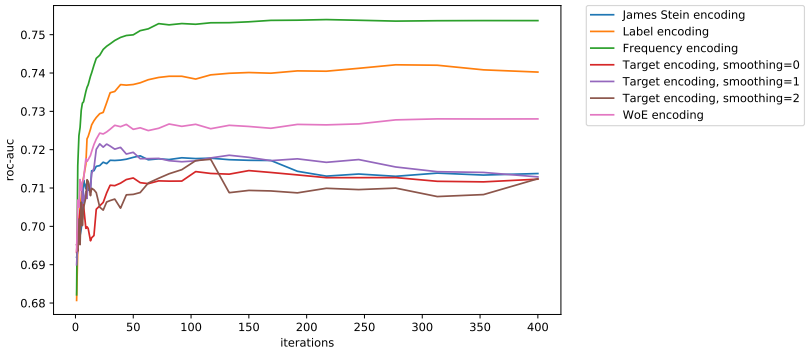
6 Эксперименты

В каждом эксперименте все категориальные признаки из набора данных кодируются одинаково одним из описанных выше методов.

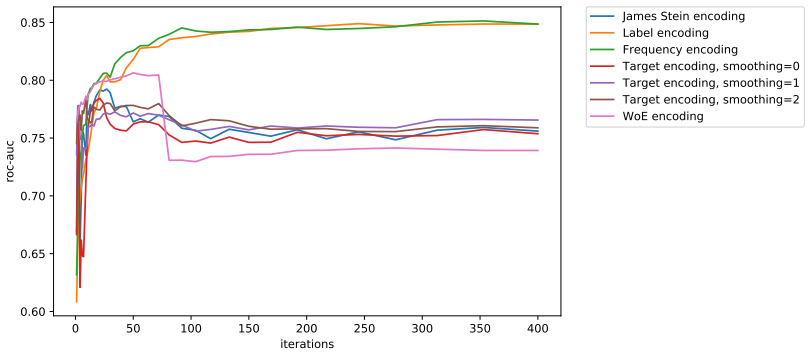
6.1 Gradient Boosting

В первую очередь были проведены эксперименты с градиентным бустингом. Для каждого набора данных и для каждого метода кодирования получалось несколько результатов на тестовом наборе при различном выборе параметра n_estimators (iterations), который отвечает за количество деревьев в модели.

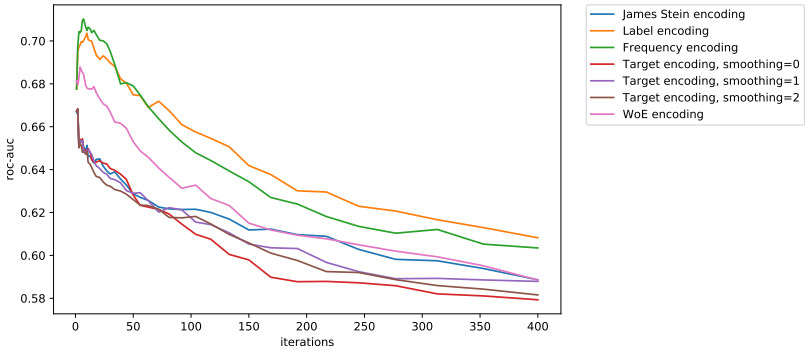
Ниже предоставлены графики зависимости качества предсказания на тестовой подвыборке для всех наборов данных.



Зависимость качества предсказания на тестовой подвыборке от iterations. График для данных Click.



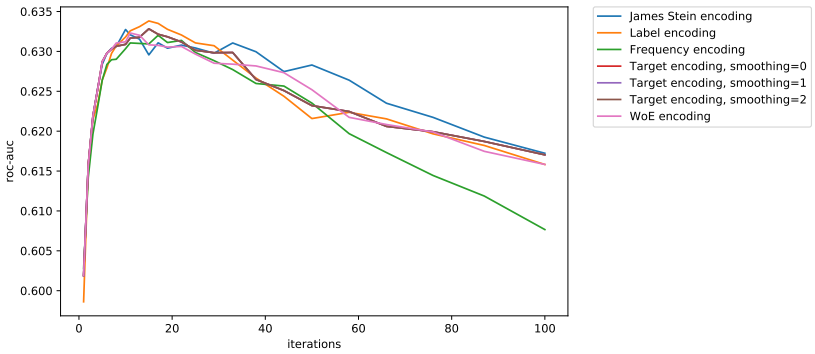
Зависимость качества предсказания на тестовой подвыборке от iterations. График для данных Amazon.



Зависимость качества предсказания на тестовой подвыборке от iterations. График для данных OneTwoTrip.

Графики для разных наборов данных получились совершенно разные: для набора Drive все методы кодирования ведут себя примерно одинаково, лучшие результаты для каждого метода незначительно разнятся, тогда как для набора Click или Amazon есть явно лучшие методы.

Ниже предоставлены лучшие результаты на тестовом наборе для каждого метода.



Зависимость качества предсказания на тестовой подвыборке от iterations. График для данных Drive.

	Click	Amazon	OneTwoTrip	Driver	Среднее
James Stein	0.7184	0.7924	0.6679	0.6327	0.7029
Label	0.7421	0.8490	0.7103	**0.6338**	0.7321
Frequency	**0.7539**	**0.8513**	**0.7103**	0.6320	**0.7368**
Target, smoothing=0	0.7145	0.7844	0.6683	0.6328	0.7000
Target, smoothing=1	0.7215	0.7782	0.6683	0.6328	0.7002
Target, smoothing=2	0.7175	0.7846	0.6682	0.6328	0.7008
WoE	0.7280	0.8064	0.6879	0.6323	0.7136

Таблица 1: Результаты работы градиентного бустинга (ROC-AUC)

При экспериментах с Target Encoding были испробованы различные значения для параметра smoothing (0, 1 и 2). На наборах Click и Amazon получались различные значения (на Click лучшим параметром оказался 1, а на Amazon 2), тогда как на наборах Drive и OneTwoTrip при различных значениях smoothing результаты получались одинаковыми.

Так же можно заметить, что Frequency encoder для всех наборов показывает один из лучших результатов. На трех наборах данных Frequency encoder лучше остальных методов, на наборе Driver незначительно проигрывает Label encoder.

6.2 Logistic regression

Аналогичные эксперименты были проведены с логичтической регрессией, результаты в таблице ниже.

	Click	Amazon	OneTwoTrip	Driver	Среднее
James Stein	0.7176	**0.8104**	0.6786	0.6100	0.7041
Label	0.6404	0.5750	0.6994	0.6218	0.6342
Frequency	0.6674	0.5738	**0.7061**	0.6303	0.6444
Target, smoothing=0	0.7339	0.7532	0.6995	0.6268	0.7033
Target, smoothing=1	0.7356	0.7512	0.6997	0.6283	0.7037
Target, smoothing=2	0.7354	0.7462	0.7002	0.6275	0.7023
WoE	**0.7382**	0.8041	0.6770	**0.6307**	**0.7125**

Таблица 2: Результаты работы логистической регрессии (ROC-AUC)

Из всех методов в среднем лучшим оказался Weight of Evidence encoder. Так как все кодирование в этом методе производится в логарифмической шкале, то Weight of Evidence encoder хорошо подходит для логистической регрессии, что доказывают эксперименты.

James Stein encoder и Target encoder показали примерно одинаковый неплохой результат, сравнимый с градиентным бустингом. Label encoder и Frequency encoder же оказались неподходящими для модели логистической регрессии: они показали очень плохой результат.

7 Заключение

На данный момент не существует единственного лучшего метода кодирования категориальных признаков, который подходит для всех задач и моделей. В настоящей работе был проведен обзор некоторых популярных подходов, а так же сравнены эффективности их работы на реальных данных.

Стоит отметить, что в работе уделялось внимание только методам кодирования по отдельности и не рассматривались объединения методов. Известно, что подобные техники позволяют сильно улучшать итоговое качество работы.

Резюмируя все эксперименты, можно сказать, что из предложенных методов на данных наборах данных для градиентного бустинга лучшим оказался Frequency encoder, для логистической регрессии - Weight of Evidence encoder