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 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 LeaveOneOut encoder

Данная кодировка схожа с Target encoder без регулярицации, для каждого объекта берется среднее арифметическое целевых переменных всех объектов с таким же признаком, но рассматриваемый объект при этом исключается. Благодаря этому достигается небольшая регуляризация.

3.6 CatBoost encoder

Данная кодировка схожа с LeaveOneOut encoder, только при кодировании объекта исключается не только он сам, но и все еще не закодированные объекты.

3.7 Weight of Evidence encoder

Данный метод кодирования применяется только для бинарного таргета. 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:

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

3.8 Probability Ratio encoder

Данный метод схож с Weight of Evidence encoder. Пусть N_+ и N_- означают то же самое, что и в описании Weight of Evidence encoder. Тогда категориальный признак кодируется как

$$\frac{N_+}{N_-}$$

Однако в случае отсутствия элементов с меткой '0' вместо N_- подставляется маленькое число, к примеру, 0.000001. Итоговая запись имеет вид:

$$\frac{N_+}{max(N_-, 0.000001)}$$

3.9 Binary encoder

В данной кодировке каждый признак кодируется бинарныи числом. Каждая цифра в записи бинарного числа будет являться признаком. Если имеется N уникальных категорий, то получим $\lceil log_2(N) \rceil$ признаков.

3.10 Helmert encoder

В данной кодировке каждое уникальное значение признака сопоставляется еще не рассмотренным уникальным значениям, при чем при рассмотрении нового значения создается новый признак: пусть остались нерассмотренными L значений, тогда новый признак будет равен $\frac{L-1}{L}$ для рассматриваемого значения и $-\frac{1}{L}$ для нерассмотренных, иначе 0. Для признака с N уникальными значениями создается N-1 новых признаков, так что для признаков с очень большим количеством уникальных значений признаковое пространство очень сильно расширяется.

3.11 Backward Difference encoder

Данная кодировка схожа с Helmert encoder. Пусть в категориальном признаке L уникальных значений, создается L-1 новых признаков. В k-ом признаке все рассмотренные уникальные значения кодируется в $-\frac{L-k}{L}$, а еще не рассмотренные в $\frac{k}{L}$.

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 200 тысяч объектов, из них классу '1' принадлежат $\approx 17.4\%$ объектов. На 160 тысячах производилось обучение, на 40 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, все 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.

5.5 Описание

Ниже предоставлено краткое описание количества уникальных значений признаков в наборах данных.

	Click	Amazon	OneTwoTrip	Driver
Всего категориальных признаков	22	9	40	50
Минимальное кол-во уникальных значений в признаке	2	67	2	2
Максимальное кол-во уникальных значений в признаке	76668	7518	812	104
Среднее кол-во уникальных значений в признаке	4507	1736	100	9
Суммарное кол-во уникальных значений в признаках	99149	15626	4005	440

Таблица 1: Описание категориальных данных в датасетах

Так как в наборе данных Click суммарное количество уникальных признаков велико, а при кодировании Helmert encoder и Backward Difference encoder признаковое пространство расширяется пропорционально этому числу, то в экспериментах на наборе Click эти два метода использоваться не будут (при применении этих методов получится почти квадратная матрица

объектов-признаков, из которой очень сложно и долго получать закономерности).

6 Эксперименты

В каждом эксперименте все категориальные признаки из набора данных кодируются одинаково одним из описанных выше методов. Так же, для избежания утечки целевой переменной и переобучения, во всех методах, в которых участвуют целевые переменные, кроме LeaveOneOut и CatBoost, применялась k-fold кодировка: для обучения моделей тренировочная выборка делилась на 6 частей, каждая часть кодировалась с помощью кодировщика, обученного на остальных 5 частях; для тестовой выборки кодировщик обучался на всех тренировочных данных. (В данные методы кодирования попадают: Target encoder, James Stein encoder, Weight of Evidence encoder, Probability Ratio encoder; LeaveOneOut и CatBoost имеют свою защиту от утечки целевой переменной)

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. Перебор параметра n_estimators осуществлялся по логарифмической сетке.

Ниже предоставлены лучшие результаты на тестовом наборе для каждого метода.

	Click	Amazon	OneTwoTrip	Driver	Среднее 1	Среднее 2
James Stein	0.7752	0.8540	0.7018	**0.6344**	0.7414	0.7301
Label	0.7447	0.8327	0.7019	0.6336	0.7282	0.7227
Frequency	0.7538	0.8340	0.7084	0.6301	0.7316	0.7242
Target, smoothing=0	0.7695	0.8427	0.7043	0.6339	0.7376	0.7270
Target, smoothing=1	0.7784	0.8502	0.6987	0.6306	0.7395	0.7265
Target, smoothing=10	**0.7804**	0.8545	0.7011	0.6323	0.7421	0.7293
WoE	0.7799	0.8527	0.7062	0.6320	0.7427	0.7303
Binary	0.7412	0.8301	0.6920	0.6296	0.7232	0.7172
Helmert	-	0.8467	0.6941	0.6291	-	0.7233
Backward Difference	-	0.7849	0.6943	0.6315	-	0.7036
Probability Ratio	0.7658	0.8457	0.7057	0.6321	0.7373	0.7278
LeaveOneOut	0.6268	0.6486	0.5434	0.5001	0.5797	0.5640
CatBoost	0.7746	**0.8551**	**0.7101**	0.6313	**0.7428**	**0.7322**

Таблица 2: Результаты работы градиентного бустинга (ROC-AUC)

В таблице колонка "Среднее 1"является средним показателем по набо-

рам данных Amazon, OneTwoTrip и Driver, "Среднее 2"является средним для всех наборов.

При экспериментах с Target Encoding были испробованы различные значения для параметра smoothing (0, 1 и 10). Значение 0 означает отсутствие регуляризации (в качестве кодирующего значения выбирается среднее по данному значению признака, без учета общего среднего). В экспериментах несколько лучше себя показало значение 10, так же эта кодировка на всех наборах показала лучшие или одни из лучших значений.

Так же стоит отметить кодировки James Stain, Weight of Evidence и CatBoost, которые показали лучшие результаты. Все остальные показали примерно одинаковые результаты, за исключением Backward Difference, Binary и LeaveOneOut кодировок - они оказались худшими для данных наборов.

Так же стоит отметить, что в пятерку лучших по сумме результатов попали 5 кодировок, имеющие доступ к целевой переменной.

6.2 Logistic regression

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

	Click	Amazon	OneTwoTrip	Driver	Среднее
James Stein	**0.7600**	0.8344	0.6966	0.6100	0.7253
Label	0.6408	0.5750	0.6994	0.5780	0.6233
Frequency	0.6697	0.5920	0.6986	0.6222	0.6456
Target, smoothing=0	0.6489	0.8092	0.7011	0.5868	0.6865
Target, smoothing=1	0.6517	0.8099	0.7012	0.5858	0.6972
Target, smoothing=10	0.6429	0.8179	0.7015	0.5767	0.6848
WoE	0.7576	0.8278	**0.7098**	0.5979	0.7233
Binary	0.7017	0.6410	0.7000	**0.6273**	0.6675
Probability Ratio	0.6516	0.7964	0.6358	0.6256	0.6774
LeaveOneOut	0.7591	0.8435	0.7030	0.6252	0.7327
CatBoost	0.7590	**0.8461**	0.7027	0.6234	**0.7328**

Таблица 3: Результаты работы логистической регрессии (ROC-AUC)

Из всех методов в среднем лучшим оказался James Stein encoder, примерно такие же результаты показал 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.

Все другие методы показали результаты в среднем сильно хуже, особенно Label encoder. Так как в этом способе кодирования задается случайный порядок между категориями, который логистическая регрессия пытается использовать, то данный метод не подходит для этой модели.

Если не брать во внимание набор Click, то все Target encoder-ы показали хорошие результаты, сильно не отличающиеся от кодировок James Stein и Weight of Evidence.

6.3 Сравнение

Ниже предоставлена таблица разности результатов бустинга и логистической регрессии.

	Click	Amazon	OneTwoTrip	Driver	Среднее
James Stein	0.0152	0.0196	0.0052	0.0244	0.0161
Label	0.1039	0.2577	0.0025	0.556	0.1049
Frequency	0.0841	0.2420	0.0098	0.0079	0.0860
Target, smoothing=0	0.1206	0.0335	0.0032	0.0471	0.0511
Target, smoothing=1	0.1267	0.0403	**-0.0025**	0.0448	0.0423
Target, smoothing=10	0.1375	0.0366	**-0.0004**	0.0556	0.0573
WoE	0.0223	0.0249	**-0.0036**	0.0341	0.0194
Binary	0.0395	0.1891	**-0.0080**	0.0023	0.0557
Probability Ratio	0.1142	0.0493	0.0699	0.0065	0.0599
LeaveOneOut	**-0.1323**	**-0.1949**	**-0.1596**	**-0.1251**	**-0.1687**
CatBoost	0.0156	-0.0090	0.0074	0.0079	0.0100

Таблица 4: Разность результатов бустинга и логистической регрессии

Заметно, что бустинг справился с задачей гораздо лучше на всех наборах данных, кроме OneTwoTrip, где обе модели спровились примерно одинаково, а так же LeaveOneOut, где бустинг показал очень плохие результаты. Так же из этой таблицы можно сделать выводы, что такие кодировщики как Label, Frequency, Binary и Probability Ratio - не лучший выбор для логистической регрессии.

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 CatBoost, несколько хуже Weight of Evidence и James Stein. Для логи-

стической регрессии лучшими оказались CatBoost и LeaveOneOut, несколько хуже Weight of Evidence и James Stein.