

Две задачи из ML

Шевляков Артём

ИМ СО РАН

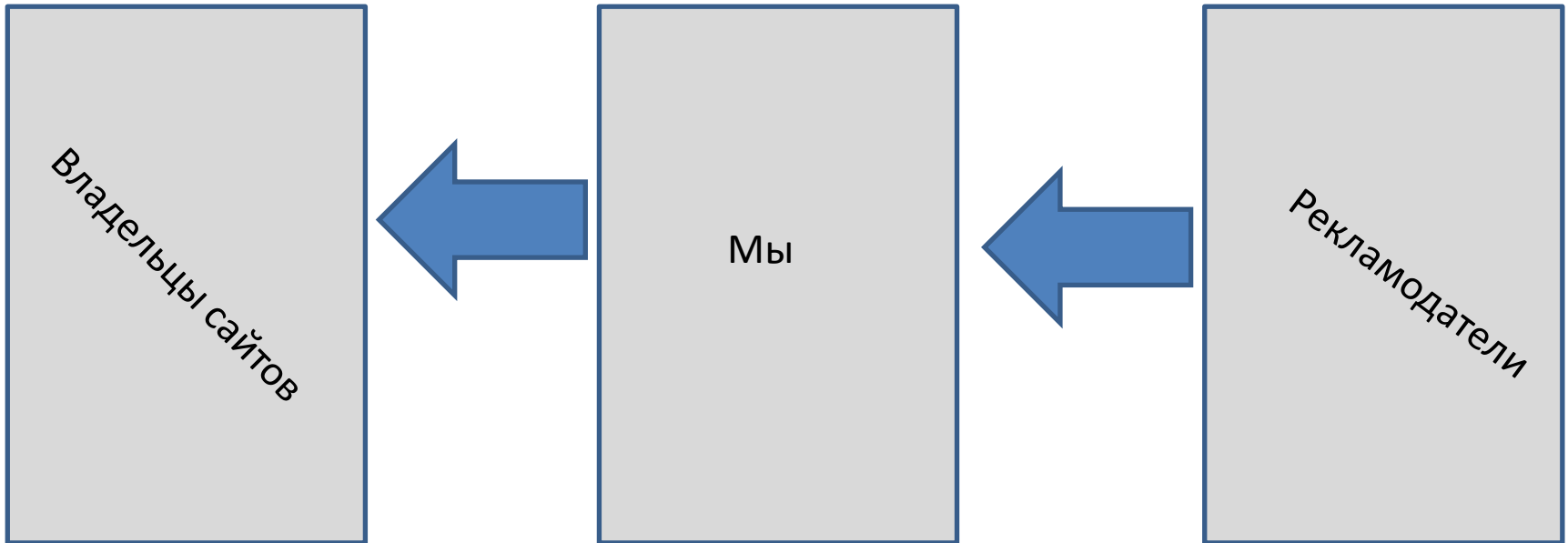
Показ рекламы на сайтах

Поиски оптимального алгоритма

Как это всё работает


Мы платим
владельцам за
каждый показ
баннера

Нам платят
рекламодатели за
каждый клик по
баннеру



VK Поиск

- Моя Страница
- Новости
- Сообщения 2
- Друзья
- Группы 1
- Фотографии
- Музыка
- Видео
- Игры 4
- Товары 7
- Документы



Отправить подарок

Подписаться

Друзья 11

Новости

Олег Елена Иван

Рамзан Сергей Александр

Интересные страницы 39

- Правительство России Государственная организация
- Московский Комсомолец (МК) СМИ
- «Эксперт» Online СМИ
- ТАСС

Ваш подарок ждет Вас
yves-rocher.ru
Сегодня за заказ Вам в подарок стильный набор из 4-х аксессуаров для хранения косметики!

Рассрочка без переплат
sovest.ru
Покупайте в рассрочку до 12 месяцев без % с помощью карты Совесть. Лимит до 300000 рублей

Сайт и зоны

Тут две зоны

Особенности задачи

- Распределение баннеров по зонам уже задано.
- Нет никакой информации о пользователях, делающих запрос на показ веб-страниц с баннерами.

Задача

На зоне два баннера. Известно, что на 1-й баннер кликают в 60% его показов, а на 2-й кликают в 40% его показов. Какая стратегия оптимальная:

- А) Первый баннер показывать 60% пользователей, а второй баннер показывать 40% пользователей.
- Б) Всегда показывать первый баннер.

Но в жизни-то всё сложнее...

- Откуда мы знаем, что 1-й баннер кликают в X% от его показов?
- Допустим каждый из баннеров был показан 10 раз. На первый кликнули 5 раз, а на второй 6 раз. Верно ли, что второй баннер лучше?
- Для корректного ответа нужно проводить много испытаний.

Решение заключается в том, что

- Наделить баннеры на зоне минимальным интеллектом и пусть они сами выяснят, кто из них самый крутой.

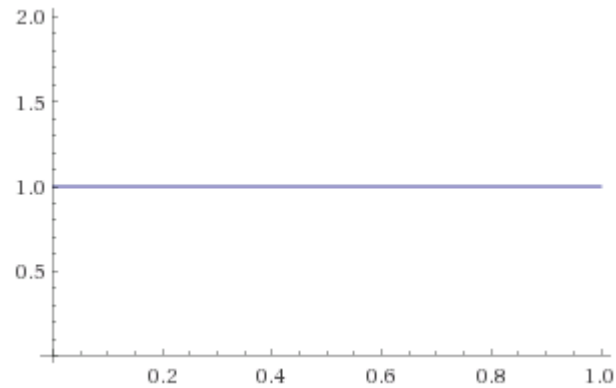


Ключевая идея

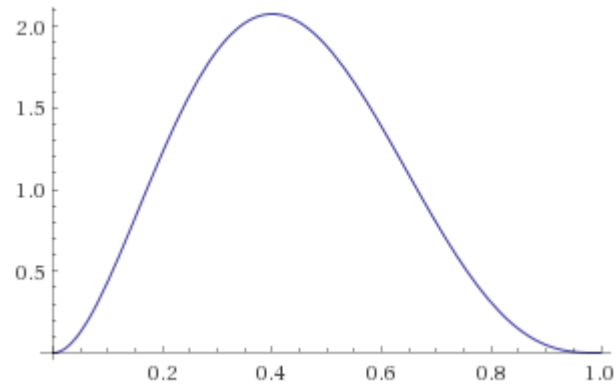
- Пусть p – вероятность клика по баннеру. Составим для нее график плотности. Высота графика выше там, где значение p более вероятно.

Например

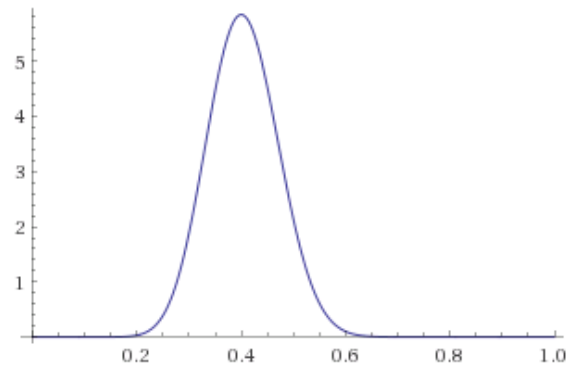
0 показов. Распределение для p :



5 показов: 2 клика+3 неудачи.
Распределение для p :



50 показов: 20 кликов+30 неудач.
Распределение для p :



Функция для графика должна:

- Быть определена на отрезке $[0,1]$.
- Иметь 2 параметра a, b (число удачных и неудачных показов).
- Иметь максимум в точке $a/(a+b)$.
- При увеличении числа показов максимум должен становиться всё выше.

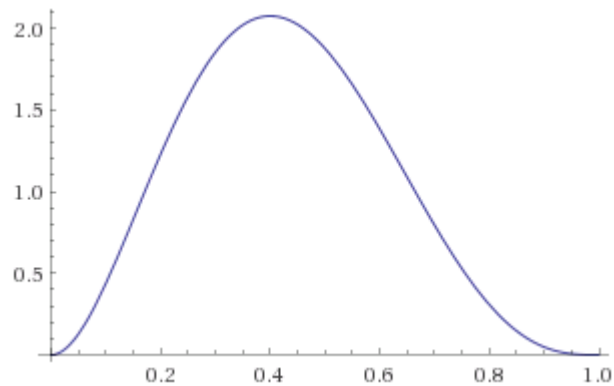
График строится на отрезке $[0,1]$ по формуле

$$\frac{x^a (1-x)^b}{B(a+1, b+1)}$$

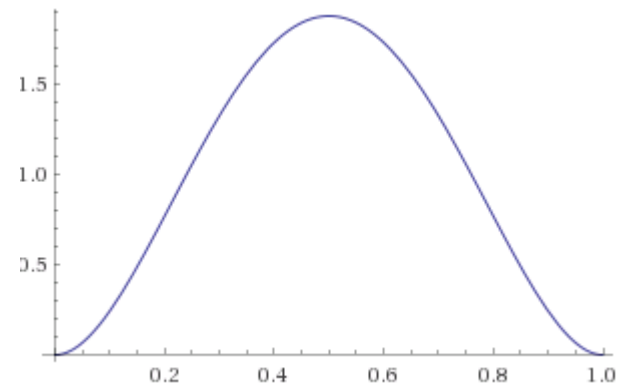
где a – число кликов, b – число неудач

Как выбрать баннер для очередного показа.

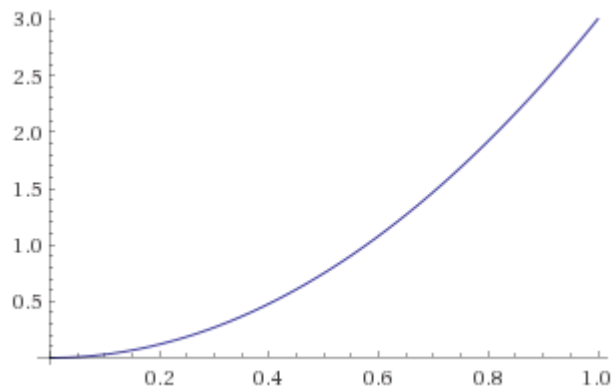
У каждого баннера своя картинка:



5 показов: 2 клика+3 неудачи.



4 показа: 2 клика+2 неудачи.



2 показа: 2 клика+0 неудач.

Thompson sampling

- Для каждого графика генерируем свое число p по принципу: чем выше график в точке x , тем выше вероятность, что будет выбран x .
- Отправить на показ баннер с максимальным числом p .
- В зависимости от результатов показа (был клик или нет) перерисовать график.

Предсказание поведения игрока в онлайн-игре

Как я работал на ферме

Игра типа «Весёлый фермер»



Это - ЖАТВА !

Была таблица в реляционной БД

userID	time	eventId
3254525656757	23435454	35
5234667637658	23435455	54
...

Сделали логирование в файлы

- Файл D:\Farm\VK\27_05_2017

userID	time	eventId
3254525656757	23435454	35
5234667637658	23435455	54
...

Формируем файл для каждого
игрока

D:\Users\3254525656757.dat
(записываем туда лишь
сознательные действия)

time	eventID
23435454	34
23435555	56
23435655	79
...	...

Из файлов игроков вычлняем признаки

userId	dur0	dur-1	dur-2	dur-3	dur-4	nAction	money Spent	plants Grown
5432656563	435	0	0	567	877	65	0	76
7967322545	0	231	0	76	0	21	13	4
...

Но есть проблема! Если одному игроку соответствует одна строка в таблице, то будет мало данных!

Один игрок даёт несколько записей

	01.01	02.01	03.01	04.01	05.01	06.01	07.01	08.01	09.01
duration	100	200	300	400	500	600	700	800	900

В таблицу добавляем несколько записей

userId	dur0	dur-1	dur-2	dur-3	dur-4	nAction	money Spent	plants Grown
5432656563	500	400	300	200	100	65	30	40
5432656563	600	500	400	300	200	21	25	50
5432656563	700	600	500	400	300	40	30	40
5432656563	800	700	600	500	400	35	15	30
5432656563	900	800	700	600	500	50	35	40

Что мы предсказывали?

- Будет ли играть на следующей неделе?
- Был ли текущий логин последним (предпоследним)?
- Будет ли игрок платить?
- Кто из заплативших игроков захочет повторить платеж?
- ...

Был ли это последний вход в игру?

userId	dur 0	dur- 1	dur- 2	dur- 3	dur-4	nAction	money Spent	plants Grown	last Login
5432656563	500	400	300	200	100	65	30	40	0
5432656563	600	500	400	300	200	21	25	50	0
5432656563	700	600	500	400	300	40	30	40	0
5432656563	800	700	600	500	400	35	15	30	0
5432656563	900	800	700	600	500	50	35	40	1
1324356557	100	0	0	0	0	34	80	10	1
9876534674	200	100	0	0	500	54	76	87	0

А далее все по шаблону ML

- Разбиваем на тренировочную и проверочную выборку.
- По тренировочной выборке мы строим модель.
- Построенную модель мы проверяем на объектах тестовой выборки и сравниваем ответ модели и точный ответ.

Секрет разбиения на тренировочную и проверочную выборки

	userId	dur 0	dur- 1	dur- 2	dur- 3	dur-4	nAction	money Spent	plants Grown	last Login
★	5432656563	500	400	300	200	100	65	30	40	0
★	5432656563	600	500	400	300	200	21	25	50	0
	5432656563	700	600	500	400	300	40	30	40	0
★	5432656563	800	700	600	500	400	35	15	30	0
★	5432656563	900	800	700	600	500	50	35	40	1
★	1324356557	100	0	0	0	0	34	80	10	1
	9876534674	200	100	0	0	500	54	76	87	0

★ - объект пошел в тренировочную выборку

Но это противостоит: модель строится по событиям, которые произошли позднее, чем события в проверочной выборке.

Что нужно делать

- Разбивать на тренировочную и проверочные выборке по дате: всё, что произошло до даты X идет в тренировочную выборку, всё остальное – в проверочную.

Проблемы

- В выборке большой перекосяк. Например, очень много неплатящих и мало платящих.
- Это и определяет поведение всех моделей ML.

Методы решения

- Удалить некоторое количество объектов преобладающего класса.
- Отклонировать некоторое количество объектов меньшего класса.
- Создать новые синтетические объекты из объектов меньшего класса (см. ниже).

- Есть два объекта из меньшего класса

userId	dur 0	dur- 1	dur- 2	dur- 3	dur-4	nAction	money Spent	plants Grown	last Login
5432656563	900	800	700	600	500	50	35	40	1
1324356557	100	0	0	0	0	34	80	10	1

- Генерируем случайное число $a \in [0,1]$.
- Новый объект будет линейной комбинацией с коэффициентами a и $(1-a)$.

Иными словами,

userId	dur 0	dur- 1	dur- 2	dur- 3	dur-4	nAction	money Spent	plants Grown	last Login
5432656563	900	800	700	600	500	50	35	40	1
1324356557	100	0	0	0	0	34	80	10	1

При $a=0.5$ будет

userId	dur 0	dur- 1	dur- 2	dur- 3	dur-4	nAction	money Spent	plants Grown	last Login
5432656563	900	800	700	600	500	50	35	40	1
1324356557	100	0	0	0	0	34	80	10	1
0000000000	500	400	350	300	250	42	57.5	25	1