Стохастический анализ неоднородно структурированных данных

Родигин Юрий Олегович, гр. 14.Б02-мм

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Вычислительная стохастика и статистические модели

Научный руководитель: к.ф.-м.н. П. В. Шпилёв
Рецензент: к.ф.-м.н. Е. Н. Мосягина

Санкт-Петербург
2018г.

Введение

В данной работе исследовались различные подходы к прогнозированию вероятностей того, что пользователь совершит переход по рекламной ссылке в интернете (англ: «Click-Through Rate Prediction»), при условии, что все переходы не случайны.

Рассматривать данную задачу можно, как **задачу бинарной классификации**.

Актуальность:

- Объем рынка интернет-рекламы за 2017г — 166,3 млрд рублей.
- Развитие систем систем собора данных о пользователях.

Работа состоит из двух частей:

- Применение готовых алгоритмов из пакета Scikit-learn на языке программирования Python.
- Адаптация методов стохастической оптимизации для задачи прогнозирования при условии, что данные поступают в режиме реального времени.

Формальная постановка задачи

Пусть задана обучающая выборка $(\boldsymbol{x}_i, y_i)_{i=1}^N$, где

$\boldsymbol{x}_i$ — вектор независимых переменных на i-ом наблюдении,

y_i — значение целевой переменной(класса) на i-ом наблюдении принимающее значение «1» или «0» (пользователь совершил переход по ссылке или нет, соответственно).

Требуется обучить алгоритмы предсказывать вероятности $p(\boldsymbol{x}_i)$ принадлежности наблюдения i к классу «1».

Scikit-learn

Scikit-learn (ранее scikits.learn) — библиотека для машинного обучения на языке программирования Python.
sklearn.preprocessing — модуль, содержащий набор классов для преобразования данных.

1. Класс **LabelEncoder** — перевод категориальных признаков в числовые значения.
2. Класс **StandardScaler** — нормализация данных

Методы:

- K-ближайших соседей — простейший метрический классификатор.
- Наивный байесовски классификатор — простой вероятностный классификатор, основанный на применении теоремы Байеса.
- Линейный дискриминантный анализ и Логистическая регрессия — ленейные классификаторы.
- Рандомизированные леса, Экстремально рандомизированные леса и Градиентный бустинг — ансамбли деревьев.

Результаты Sklearn

- Объем выборки: 10 дней — 100 000 записей
- Обучение — первые 9 дней (90 000 записей), валидация — последний день (10 000 записей).

$$LogLoss = \frac{1}{N}\sum_{j=1}^{N}[-y_j \log p(\boldsymbol{x}_j) - (1 - y_j)\log(1 - p(\boldsymbol{x}_j))],$$

Таблица: Ошибка $LogLoss$ на валидации при применении SKlearn

Алгоритм	Cross validation $LogLoss$
Наивный байесовский классификатор	0.5643
Рандомизированные леса	0.5528
Экстремально рандомизированные леса	0.5427
K-ближайших соседей	0.4526
Логистическая регрессия	0.4465
Линейный дискриминантный анализ	0.4400
Градиентный бустинг	0.4170

Данные поступают потоком в реальном времени.

Проблема: LabelEncoder и StandardScaler требуется проход по всей выборке перед преобразованиями.

One-hot encoding: преобразование текстовых характеристик в матрицу $N \times M$ со значениями «0» и «1», где N — это кол-во записей, а M — кол-во уникальных значений у этой характеристики.

Таблица: Пример one-hot encoding

Исходное знач.	Преобразованное
A	1 0 0
B	0 1 0
C	0 0 1
A	1 0 0

Feature hashing

При большом количестве категориальных признаков размерность матрицы one-hot encoding начинает быстро расти.

Решение: **hashing trick**, предоставляющий хэш-функции:

feat = header[m] + '_' + feat
feat = abs(hash(feat)) % D — ограничение по знаку и длине

Чтобы разделить одинаковые значения разных признаков хэшировались пары **название признака** + **значение признака**.

Основные достоинства:

- Сохраняется максимум информации
- Быстрая работа с пространствами с большим количеством характеристик.

Преимущества:

- Простота реализации,
- Скорость обучения,
- Качество предсказания.

y — зависимая переменная принимающая значение «1» или «0»,

$\boldsymbol{x}$ — вектор независимых переменных,

$\boldsymbol{w} = (w_1, ..., w_n)$ —вектор параметров (коэффициентов регрессии).

$z(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = w_1 x_1 + \ldots + w_n x_n$,

Логистическая функция (сигмоид):

$$\mathbb{P}\{y = 1 \mid \boldsymbol{x}\} = \frac{1}{1 + e^{-z(\boldsymbol{x})}},$$

значение которой является апостериорной вероятностью принадлежности наблюдения к классу «1».

Обучение алгоритма

Для настройки параметров (весов) использовались:

$g_j \longleftarrow \frac{\partial NLL}{\partial w} = p_j - y_j$ — градиент на j-ой записи,

$$NLL = -y_t \log p_t - (1 - y_t) \log(1 - p_t).$$

- **Стохастический градиентный спуск (SGD)**:

$$w_i \longleftarrow w_i - \alpha g_j,$$

где w_i — вес i-го признака, α — темп обучения (learning rate),

- **Адаптивный градиентный спуск (AdaGrad)**:

$$w_i \longleftarrow w_i - \frac{\alpha}{\sqrt{n_i}} g_j,$$

где $n_i \longleftarrow n_i + g_j^2$ — сумма квадратов градиентов.

Результаты логистической регрессии

Входные параметры:

- $\alpha = 0.001$
- количество эпох (проходов по обучающей выборке) — 6

При обучении с использованием только AdaGrad получена ошибка на валидации — 0.4002.

При обучении с использованием только SGD получена ошибка на валидации — 0.3957.

При комбинации SGD и AdaGrad: первые 3 эпохи с SGD и следующие с AdaGrad — 0.3951.

График ошибки на обучении

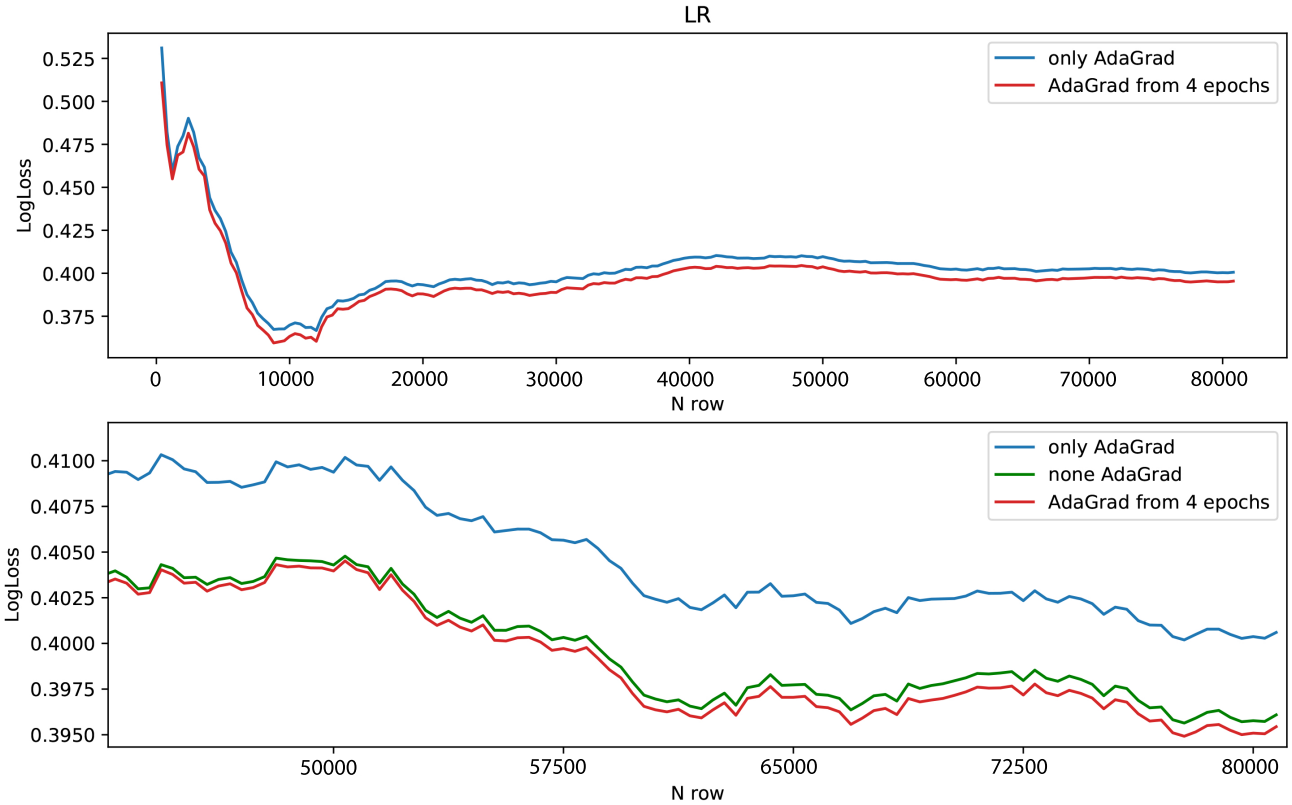

Рис.: 2 Ошибка на обучении

N row — число записей для обучения.

$$LogLoss = \frac{1}{N}\sum_{j=1}^{N}[-y_j \log p(\boldsymbol{x}_j) - (1-y_j)\log(1-p(\boldsymbol{x}_j))],$$

Предположение: Целевая переменная может зависеть от парных взаимодействий между признаками.

Решение: Полиномиальная регрессия второго порядка:

$$\mathbb{P}\{y = 1 \mid \boldsymbol{x}\} = \frac{1}{1 + e^{-z(\boldsymbol{x})}}, \qquad z(\boldsymbol{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} w_{ij} x_i x_j,$$

Проблема: Модель состоит из $d(d-1)/2 + d + 1$ параметров.

Вес взаимодействия признаков i и j может быть аппроксимирован произведением низкоразмерных скрытых векторов $\boldsymbol{v}_i$ и $\boldsymbol{v}_j$.

Это дает нам модель, называемую **Факторизационной машиной(FM)**:

$$z(\boldsymbol{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_j,$$

где вектора $\boldsymbol{v}_i, \boldsymbol{v}_j$ имеют размерность r, **задаваемую вручную**.

Число параметров снижается до $dr + d + 1$.

- Follow the Regularized Leader (FTRL)
- L1 и L2 регуляризация

$i \in \{1, \ldots, D\}$, D – размерность признакового пространства,
$t \in \{1, \ldots, N\}$, N – кол-во тренировочных записей.

$$w_{t,i} \longleftarrow \begin{cases} 0, & \text{если} |z_i| \leq \lambda_1; \\ -\left(\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda_2\right)^{-1} (z_i - \text{sign}(z_i)\lambda_1), & \text{иначе.} \end{cases}$$

$$g_t \leftarrow p_t - y_t.$$

$$\sigma_i \leftarrow \frac{1}{\alpha}\left(\sqrt{n_i + g_t^2} - \sqrt{n_i}\right),$$

$$z_i \leftarrow z_i + g_t - \sigma_i w_{t,i},$$

$$n_i \leftarrow n_i + g_t^2,$$

Коэффициенты $v_{t,i,k}$, $k \in \{1, \ldots, r\}$, r — размерность скрытых векторов, вычисляются аналогично, за исключением градиента:

$$g_t^{fm} \longleftarrow g_t s_{i,k},$$

где

$$s_{i,k} \longleftarrow s_{i,k} + \sum_{j \neq i} \sum_{k}^{r} v_{j,k}.$$

Результаты факторизационных машин

$$z(\boldsymbol{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_j,$$

Входные параметры:

- Количество эпох — 6,
- $r = 4$ — размерность скрытых векторов.
- Для весов $\boldsymbol{w}$:
  - $\alpha = 0.1$, $\beta = 1$ — параметры скорости обучения для w_i,
  - $\lambda_1 = 1$ и $\lambda_2 = 0.1$ — параметры регуляризации.
- Для весов $\boldsymbol{v}$:
  - $\alpha_{fm} = 0.05$ и $\beta_{fm} = 1$ — параметры скорости обучения,
  - $\lambda_{1_{fm}} = 2$ и $\lambda_{2_{fm}} = 1$ — параметр регуляризации.

$$LogLoss = \frac{1}{N} \sum_{j=1}^{N} [-y_j \log p(\boldsymbol{x}_j) - (1 - y_j) \log(1 - p(\boldsymbol{x}_j))],$$

Ошибка $LogLoss$ на валидационной выборке — 0.4062

График ошибки на обучении

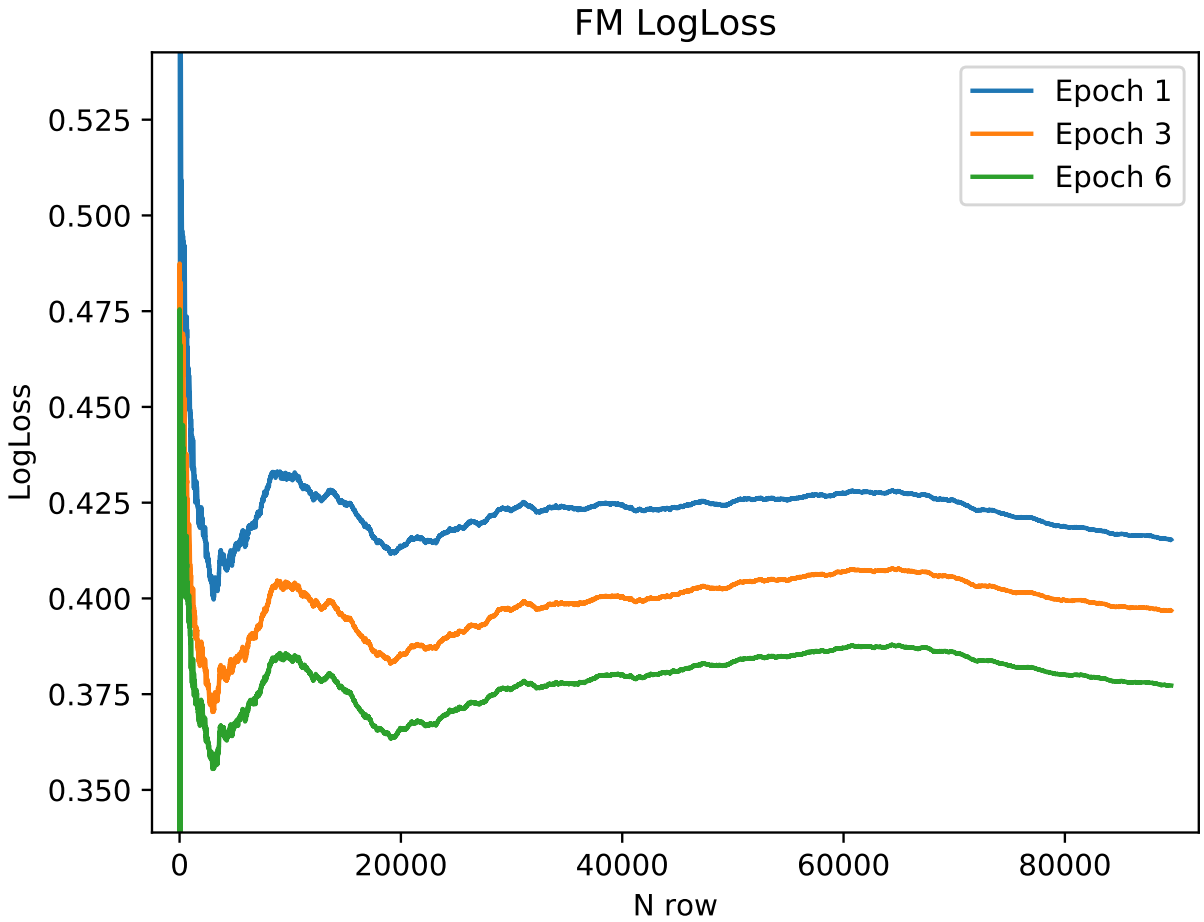

Рис.: 3 Ошибка на обучении

N row — число записей для обучения

$$LogLoss = \frac{1}{N}\sum_{j=1}^{N}[-y_j \log p(\boldsymbol{x}_j) - (1-y_j)\log(1-p(\boldsymbol{x}_j))],$$

Заключение

- Были рассмотрены и применены основные алгоритмы из библиотеки Scikit-learn, позволяющие оценивать вероятности принадлежности к классам.
- Реализован алгоритм логистической регрессии с различными модификациями, поддерживающий работу с большими объемами данных за счет онлайн-обучения.
- Реализован алгоритм факторизационных машин, допускающий эффективное использование предположения о эффекте взаимодействия признаков между собой, и так же поддерживающий онлайн-обучение.