Логистическая регрессия

для решения задач классификации

Вихляев Егор, ММТ-2

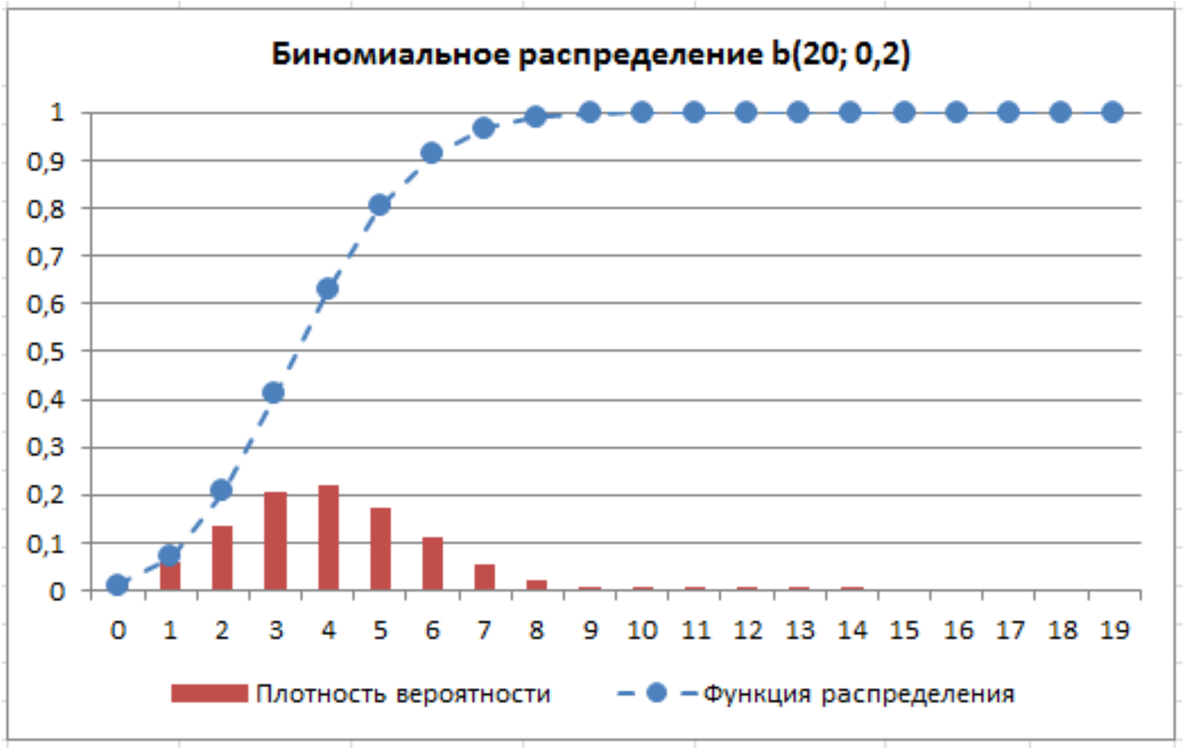
Описание проблемы

Логистическая регрессия - это статистический метод, который обычно используется для решения задач классификации, а не задач регрессии, несмотря на свое название. Вот несколько примеров проблем, для решения которых часто используется логистическая регрессия:

- Бинарная классификация, где цель - разделить объекты на два класса. Например, определение, является ли письмо спамом или не спамом, определение, болен ли пациент определенной болезнью (да/нет), и многие другие подобные задачи.
- Многоклассовая классификация, где объекты должны быть отнесены к одному из нескольких классов. Это делается с использованием множества бинарных классификаторов или методов, таких как «один против всех» или «многие против многих».
- Медицинская диагностика, где определяется наличие или отсутствие болезни на основе медицинских показателей и тестов.
- Логистическая регрессия может применяться для анализа данных о клиентах, определения, склонны ли они к покупке определенных товаров или услуг, или к каким-либо другим действиям.
- Логистическая регрессия хорошо подходит для задач, где требуется оценивать вероятность принадлежности объекта к классу, и она широко используется во многих областях, где необходима классификация и прогнозирование с учетом вероятностей.

Концептуальная постановка

На практике логистическая регрессия используется для решения задач классификации с линейно-разделяемыми классами. При этом, предполагается, что зависимая переменная принимает два значения и имеет биномиальное распределение.



Математическая постановка

Постановка задачи восстановления логистической регрессии

Пусть дана выборка (X, Y), где $X = \begin{bmatrix} X_1 \\ ... \\ X_m \end{bmatrix}$ – набор объектов, $Y = \begin{bmatrix} Y_1 \\ ... \\ Y_m \end{bmatrix}$ - ответы на этих объектах. $x_i \in \mathbb{R}^n$. $y_i \in 0; 1$ - один из двух классов: либо 0, либо 1.

Нужно построить функцию $f : x_i \rightsquigarrow \widetilde{y}_i \approx y_i$. Логистическая регрессия задает структуру данной функции.

$\widetilde{y}_i = f(x_i; w, b) = \sigma(w^T x_i + b)$ - сигмоида от суммы скалярного прозведения и смещения. $w \in \mathbb{R}^n$ - вещ. вектор, $b \in \mathbb{R}$ - вещ. число (смещение).

Математическая постановка

Постановка задачи восстановления логистической регрессии

Логистическая регрессия для объекта x_i получает предсказание в два шага:

1. $z = (-w^T-) \begin{bmatrix} | \\ x_i \\ | \end{bmatrix} + b \in \mathbb{R}$ - получаем из вектора число.
2. Полученное число преобразуем в меру активации: $a = \sigma(z) = \frac{1}{1+e^{-z}}$ - логистическая функция или сигмоида.

Таким образом мы задали модель и способ вычисления предсказания. Поскольку в способе есть два неизвестных параметра w и b, то подберем их через функцию потерь:

$$L_i(\widetilde{y_i}) = -y_i log(\widetilde{y_i}) - (1 - y_i) log(1 - \widetilde{y_i}).$$

Математическая постановка

Постановка задачи восстановления логистической регрессии

На выборке $(X, Y) : L = \frac{1}{m}\sum_{i=1}^{m} L_i(\widetilde{y}_i) \to min(w, b) = \frac{1}{m}\sum_{i=1}^{m} L_i(\sigma(w^T x_i + b)) \to min(w, b)$. Задача свелась к минимизации функции потерь. Минимизируем градиентным спуском.

$w_{j+1} := w_j - \alpha\frac{\partial L(w_j, b_j)}{\partial w}$, где w_j - параметр на текущем шаге, α - шаг градиентного спуска, $\frac{\partial L(w_j, b_j)}{\partial w}$ - градиент. Аналогично для параметра b.

Поскольку $L_i(w, b) = L_i(a(z(w, b)))$ - сложная функция, то ищем ее частную производную следующим образом:

$$\frac{\partial L(w_j, b_j)}{\partial w} = \frac{\partial L_i}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w}.$$

Математическая постановка

Постановка задачи восстановления логистической регрессии

Тогда:

$$\frac{\partial L_i}{\partial a} = -\frac{y_i}{a} + \frac{1 - y_i}{1 - a} = \frac{a - y_i}{a(1 - a)},$$

$$\frac{\partial a}{\partial z} = a(1 - a),$$

$$\frac{\partial z}{\partial w} = \frac{\partial (w^T x_i + b)}{\partial w} = \begin{bmatrix} | \\ x_i \\ | \end{bmatrix}.$$

Таким образом:

$$\frac{\partial L_i}{\partial w} = \frac{a - y_i}{a(1 - a)} * a(1 - a) * x_i = (a - y_i x_i),$$

при этом, аналогично,

$$\frac{\partial L_i}{\partial b} = a - y_i.$$

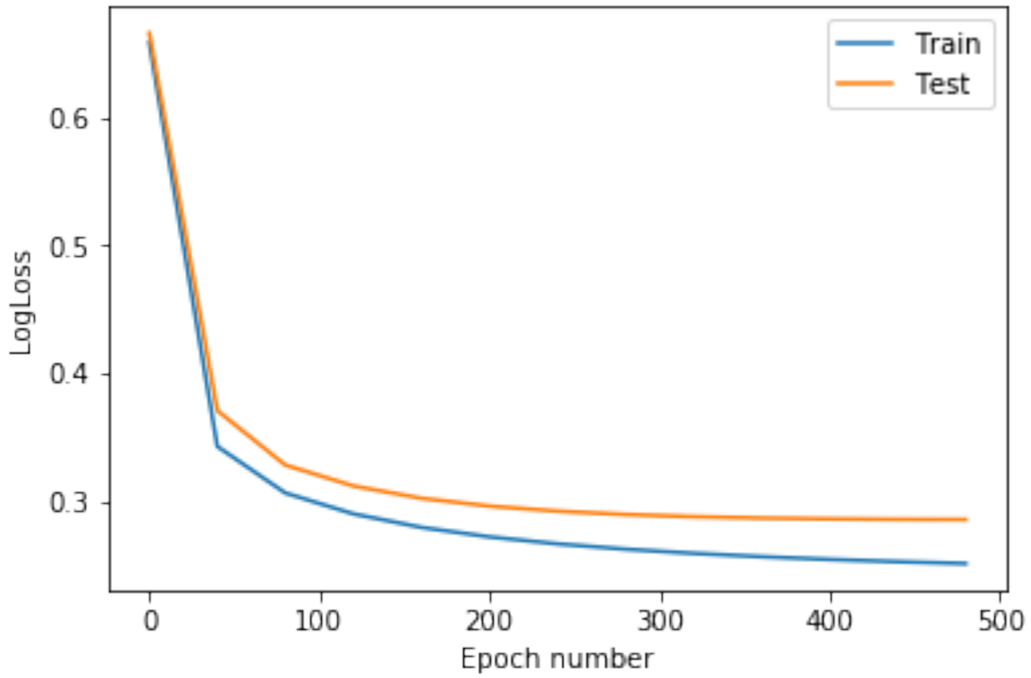
Решение

Модельные данные: выборка цифр от 0 до 9 в виде пиксельных картинок

Google Colab

Визуализация результатов

Модельные данные: выборка цифр от 0 до 9 в виде пиксельных картинок



Анализ результатов

Модельные данные: выборка цифр от 0 до 9 в виде пиксельных картинок

Алгоритм допустил около 11% ошибок классификации на тестовой выборке и порядка 9% на тренировочной выборке. Разница обусловлена тем, что модель всегда быстрее обучается на тренировочных данных.

Решение

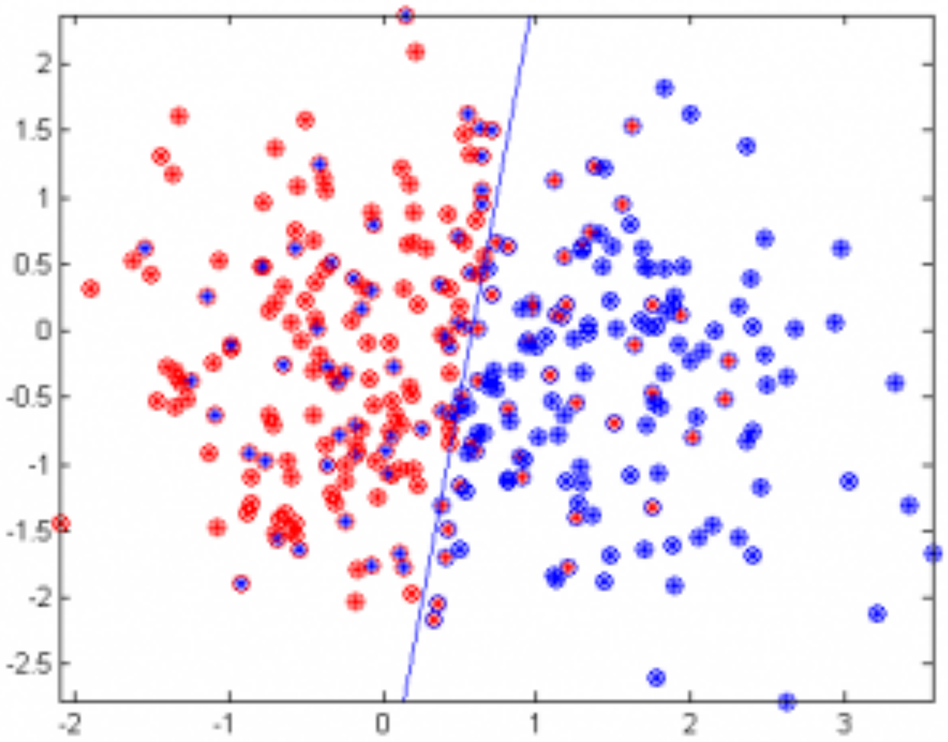
Модельные данные: сильно перекрывающиеся классы

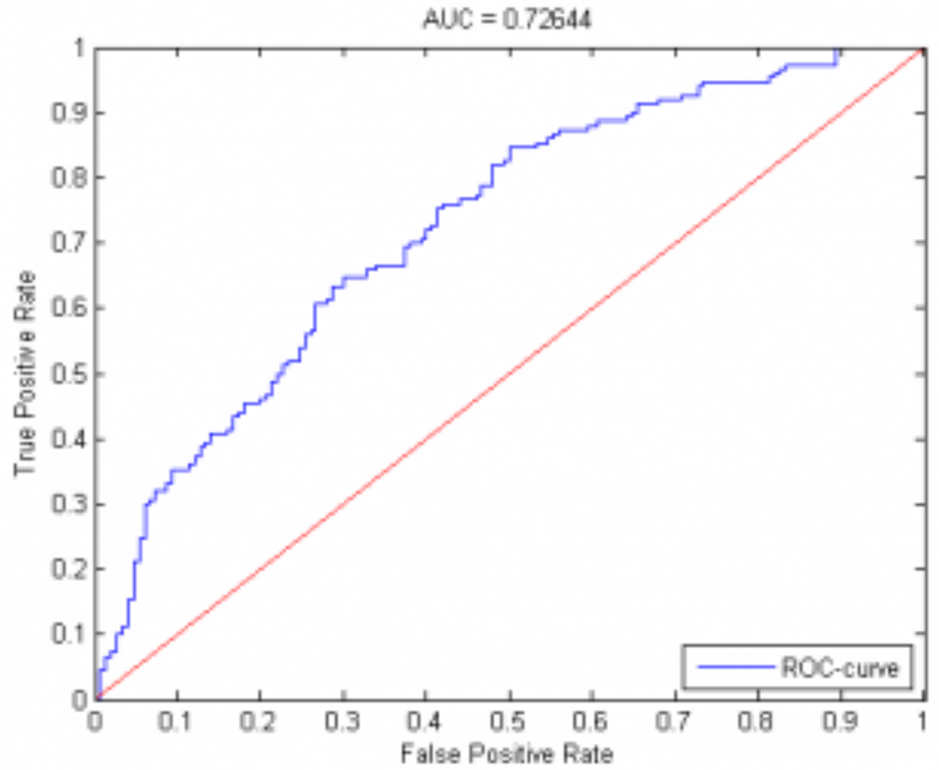
```
%% Создаем демонстрационный датасет

N = 150; % 1-й класс содержит 150 объектов
alpha = 1; % 2-й класс содержит alpha*N объектов
sig2 = 1; % предположим, что 2-й класс имеет ту же
          дисперсию, что и 1-й
dist2 = 1;

[X, y] = loadModelData(N, alpha, sig2, dist2);

%% Визуализируем входные данные
idx1 = find(y == 0); % индексы объектов для 1-го класса
idx2 = find(y == 1); % индексы объектов для 2-го класса


h = figure; hold on
plot(X(idx1,1), X(idx1,2), 'r*');
plot(X(idx2,1), X(idx2,2), 'b*');
axis tight
%close(h);
```

```
%% Тренировка датасета
w = logreglearn(X,y);

%% Перезаписываю регрессию в те же данные
p = logregmdl(w,X);

%% Выводим результат

idx1 = find(p > 1/2); % индексы объектов для 1-го класса
idx2 = find(p <= 1/2); % индексы объектов для 2-го класса
% визуализируем классифицированные данные
plot(X(idx1,1), X(idx1,2), 'bo');
plot(X(idx2,1), X(idx2,2), 'ro');

% гиперплоскость разделения классов, образованная как функция
  (У нас это линия)
separateXLim = inline( '(-b(1)- YLim*b(3))/b(2)', 'b','YLim');
% чертим нашу линию разделения
plot(separateXLim(w,ylim), ylim, ‘b-');
```

Визуализация результатов

Модельные данные: сильно перекрывающиеся классы





Анализ результатов

Модельные данные: сильно перекрывающиеся классы

Алгоритм допустил около 30% ошибок классификации. В общем и целом, именно это и следовало ожидать, поскольку входные данные были принципиально линейно неразделимы. Для этого алгоритма построена кривая ошибок на этих модельных данных, представленная на предыдущем слайде справа.

Тем не менее, для более точных результатов и более надежных прогнозов необходимо использовать достаточное количество качественных данных. Также важно учитывать ограничения и предпосылки, которые могут влиять на результаты логистической регрессии, а так же ставить четкие цели и вопросы, на которые должен дать ответ логистический регрессионный анализ.