Теоретическая часть

Постановка задачи

Есть множество объектов X (сообщений) и множество ответов Y (ответов). Будем считать, что декартово произведение $X \times Y$ (множество пар объект – ответ) образует вероятностное пространство. Пусть есть некая выборка: $X^l = (x_i, y_i)$. Требуется найти такой алгоритм $a: X \to Y$, который с минимальной ошибкой сопоставляет любому объекту $x \epsilon X$ ответ $y \epsilon Y$.

Для этого введем допущение: пусть известна плотность

$$p(x, y) = p(x)P(y|x) = P(y)p(x|y).$$

$p(x, y)$ – плотность вероятности,

$P(y)$ – априорная вероятность класса Y (без учета свойств объекта),

$p(x|y)$ – функция правдоподобия класса Y,

$P(y|x)$ – апостериорная вероятность класса Y (с учетом свойств объекта).

Принцип максимума апостериорной вероятности

$$a(x) = arg\max_{y \epsilon Y} P(y|x) = arg\max_{y \epsilon Y} P(y)p(x|y).$$

Данный классификатор называют оптимальным Байесовским классификатором.

Функционал среднего риска

Проведем анализ ошибок. Если объект был класса y, а классификатор отнес его к классу s.

Введем штрафы за классификацию (потери) λ_{ys}. Формализуем понятие мат. ожидания ошибки. Пусть $a: X \to Y$ разбивает X на непересекающиеся области:

$$A_y = \{x \epsilon X | a(x) = y\}, y \epsilon Y.$$

Вероятность ошибки $P(A_y, y) = \int_{A_s} p(x, y)dx$. Из определения мат. ожидания имеем функцию среднего риска:

$$R(a) = \sum_{y \epsilon Y} \sum_{s \epsilon Y} \lambda_{ys} P(A_s, y).$$

Теорема. Если известны $P(y)$ и $p(x|y)$, то минимальный средний риск $R(a)$ имеет Байесовский классификатор:

$$a(x) = arg \min_{s \epsilon Y} \sum_{y \epsilon Y} \lambda_{ys} P(y) p(x|y) .$$

Теорема. Если $\lambda_{yy} = 0$, $\lambda_{ys} = \lambda_y$. Для $\forall y, s \epsilon Y$, то минимум среднего риска $R(a)$ достигается при

$$a(x) = arg \max_{s \epsilon Y} \lambda_y P(y) p(x|y).$$

Наивный байесовский классификатор

Признаки объекта делаются независимыми случайными величинами.

Рассмотрим задачу классификации сообщений (выделения из них спама). На стадии обучения для каждого встреченного слова высчитывается его вес – оценка вероятности того, что письмо с этим слово является спамом. При классификации отнесение письма к спаму осуществляется путем сравнения общего веса письма с планкой, заданной пользователем. Обычно это 0.6-0.8.

Пусть есть обучающая выборка сообщений $M: W_i \epsilon M$, где W_i – все слова из выборки. Тогда вероятность того, что случайное сообщение из N слов W_i, $i = \overline{1, N}$, является спамом (S), вычисляется по формуле:

$$P = \frac{\prod_{i=1}^{N} P_i}{\prod_{i=1}^{N} P_i + \prod_{i=1}^{N} (1 - P_i)}.$$

Здесь $P = P(S|W_1 \dots W_N)$ – условная вероятность того, что сообщение, содержащее слова W_i, $i = \overline{1, N}$, является спамом. Вероятность $P_i = P(S|W_i)$ – вероятность того, что сообщение, содержащее слово W_i, – спам[1].

$$P(S|W_i) = \frac{P(W_i|S)}{P(W_i|S) + P(W_i|H)}.$$

$P(W_i|S)$ – это вероятность того, что сообщение, являющееся спамом (S), содержит слово W_i. Эта величина вычисляется по формуле:

$$P(W_i|S) = \frac{count(M: W_i \epsilon M, M \epsilon S)}{\sum_j count(M: W_j \epsilon M, M \epsilon S)}.$$

Т. е. величина $P(W_i|S)$ – это относительная частота сообщений, содержащих слово W_i, которые были отнесены к спаму во время фазы обучения.

$P(W_i|H)$ – это вероятность того, что сообщение, не являющееся спамом (H), содержит слово W_i. Эта величина вычисляется по формуле:

$$P(W_i|H) = \frac{count(M: W_i \epsilon M, M \epsilon H)}{\sum_j count(M: W_j \epsilon M, M \epsilon H)}.$$

[1] В данной формуле опущены априорные вероятности P(S) и P(H), принадлежности и не принадлежности к спаму соответственно. Классификаторы, которые используют такую формулу, называются фильтрами «без предубеждений».

Т. е. величина $P(W_i|H)$ – это относительная частота сообщений, содержащих слово W_i, которые не были отнесены к спаму (H) во время фазы обучения.

Практическая реализация

Для удобства вычисления в работе предлагается использовать следующие изменения.

1. Изменим формулу вычисления вероятности $P(S|W_1 \dots W_N)$:

$$P(S|W_1 \dots W_N) = \frac{P(W_1 \dots W_N|S)P(S)}{P(W_1 \dots W_N)} = \frac{P(S)\prod_{i=1}^{N} P'(W_i|S)}{P(W_1 \dots W_N)}.$$

2. Не требуется учитывать $P(W_1 \dots W_N)$ при вычислении $P(S|W_1 \dots W_N)$ так как это не влияет на выбор аргумента при использовании $argmax$.
3. При выборе категории будем находить $argmax$ от логарифма $P(Cat|W_1 \dots W_N)$ по Cat (категория).
4. Чтобы избежать проблемы редких слов, вероятность $P'(W_i|S)$ будем считать, как средневзвешенную вероятность:

$$P'(W_i|S) = \frac{weight \times P_a(W_i) + total \times P(W_i|S)}{weight + total}.$$

5. В практической реализации необходимо увеличить количество категорий и уметь работать с любым количеством категорий.

Практическая часть

Задание по практической реализации алгоритма находится в файле problems-1.ipynb