

Дисперсионный анализ. Метод главных компонент. Логистическая регрессия

Теория вероятностей  
и математическая статистика / Урок 8



## Дисперсионный анализ

Дисперсионный анализ — метод в математической статистике, направленный на поиск зависимостей в данных, в которых целевая переменная является *количественной*, а факторы являются *категориальными*.

В однофакторном дисперсионном анализе исследуется влияние одного категориального фактора  $x$  на переменную  $y$ . Допустим, у фактора  $x$  имеется  $k$  разных значений или *уровней*. На практике это означает, что у нас имеется  $k$  выборок:

$$Y_1, \dots, Y_k,$$

и выборка  $Y_i$  соответствует значениям переменной  $y$  на  $i$ -м уровне фактора  $x$ .

Итак, нулевая гипотеза  $H_0$  утверждает, что средние по всем этим выборкам равны:

$$H_0 : \bar{Y}_1 = \dots = \bar{Y}_k$$

Другими словами, нулевая гипотеза заключается в том, что фактор  $x$  никак не влияет на значения переменной  $y$ .

Для проверки гипотез в дисперсионном анализе используется **F-критерий Фишера**. Используемая статистика представляет из себя отношение дисперсии между уровнями к дисперсии внутри уровней.

Пусть в каждой выборке  $Y_i$  содержится  $n_i$  элементов. Обозначим через  $Y$  объединение всех выборок, т.е. выборку размера  $n = n_1 + \dots + n_k$ .

Рассмотрим две суммы квадратов:

$$SS_b = \sum_{i=1}^k (\bar{Y}_i - \bar{Y})^2 n_i, \quad SS_w = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{Y}_i)^2,$$

где  $y_{ij}$  —  $j$ -й элемент  $i$ -й выборки.

Первая сумма — отклонения между группами («b» от слова Between — между), вторая — отклонения внутри групп («w» от слова Within — внутри).

По этим значениям вычисляются соответствующие несмещённые оценки дисперсий:

$$\sigma_b^2 = \frac{SS_b}{k-1}, \quad \sigma_w^2 = \frac{SS_w}{n-k}$$

Итак, статистика для проверки гипотезы  $H_0$ :

$$F = \frac{\sigma_b^2}{\sigma_w^2}$$

В предположении верности гипотезы  $H_0$  статистика  $F$  имеет распределение Фишера с параметрами  $k_1 = k - 1$ ,  $k_2 = n - k$ . Критическая область здесь правосторонняя:

$$\Omega_\alpha = (t_{1-\alpha, k_1, k_2}, \infty),$$

где  $t_{x, k_1, k_2}$  — квантиль порядка  $x$  для распределения Фишера с параметрами  $k_1, k_2$ .

Существует также процедура **двухфакторного дисперсионного анализа** для случая, когда имеется пара категориальных факторов, и требуется исследовать влияние каждого из них на целевую переменную.

Про двухфакторный дисперсионный анализ читайте в дополнительных материалах к уроку.

## Метод главных компонент

Метод главных компонент — один из методов анализа факторов и понижения их размерности, т.е. приведения множества непосредственно наблюдаемых факторов  $x_i, i = 1, \dots, m$ , к меньшему числу новых линейно независимых факторов  $y_j, j = 1, \dots, q, q < m$ .

С помощью этого метода можно:

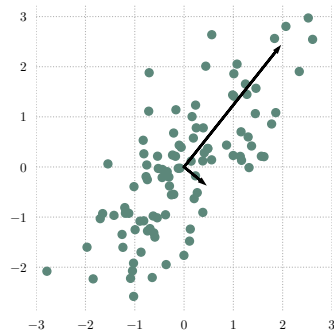
- Проанализировать уровень совокупной линейной зависимости в имеющихся данных (а не только лишь попарной, как в корреляционном анализе).
- Понизить размерность в данных и одновременно избавиться от линейной зависимости, контролируя при этом уровень потери информации.

В основе метода главных компонент лежит работа с *собственными векторами* и *собственными числами* матрицы ковариаций. В дополнительных материалах к уроку представлен краткий экскурс в эту тематику.



В контексте метода главных компонент нас интересует следующее: для матрицы ковариаций собственные векторы являются главными направлениями вариативности в данных (или главными компонентами).

Причём чем больше собственное значение, тем сильнее степень вариативности данных в этом направлении.



Итак,

- ① метод главных компонент перераспределяет вариативность внутри факторов  $x_1, \dots, x_m$  вдоль главных компонент,
- ② при этом главные компоненты с малым уровнем вариативности (т.е. с малыми собственными значениями) можно исключить.

Итак,

- 1 метод главных компонент перераспределяет вариативность внутри факторов  $x_1, \dots, x_m$  вдоль главных компонент,
- 2 при этом главные компоненты с малым уровнем вариативности (т.е. с малыми собственными значениями) можно исключить.

Допустим, имеется матрица объект-признак:  $X = (x_{ij})_{n \times m}$  (т.е.  $n$  объектов,  $m$  признаков). Предположим также, что в процессе преобразования данных мы готовы потерять не более, чем долю  $p$  от имеющейся информации (например, это может быть ограничение в 5%).

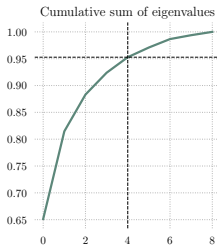
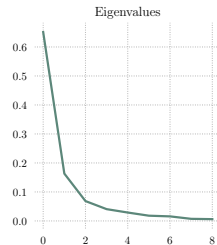
- 1 *Центрируем* матрицу  $X$ , т.е. вычтем из каждого столбца среднее по этому столбцу. В результате получится матрица  $X^* = (x_{ij}^*)_{n \times m}$ , в которой средние по столбцам равны 0. Обозначим сумму дисперсий признаков за  $\sigma_X$ , она нам ещё пригодится.
- 2 Вычислим матрицу ковариаций  $\Sigma = (\sigma_{ij})_{m \times m}$ .
- 3 Вычислим собственные векторы и собственные значения матрицы  $\Sigma$ . Пусть это собственные значения  $\lambda_1, \dots, \lambda_m$ , расположенные в порядке убывания, и им соответствуют векторы  $v_1, \dots, v_m$ . Сумма собственных значений равна  $\sigma_X$ . Собственные векторы  $v_i$  называются **главными компонентами**, и уровень вариативности данных вдоль каждой компоненты равен  $\lambda_i$ .

- ④ Найдём наименьшее  $k$ , для которого верно:

$$\frac{\lambda_1 + \dots + \lambda_k}{\sigma_X} \geq 1 - p$$

Значение слева называется **долей объяснённой дисперсии** выбранных главных компонент: в знаменателе стоит дисперсия признаков до применения метода, а в числителе стоит то, что будет дисперсией новых признаков.

- ⑤ Составим матрицу  $T$  из первых  $k$  собственных векторов (столбцов). Тогда новая матрица объект-признак:  
 $Y = X^* \cdot T$ .



# Логистическая регрессия

Логистическая регрессия возникает в задачах **бинарной классификации**: исследуется набор объектов, и каждому объекту приписана бинарная метка (0 или 1).

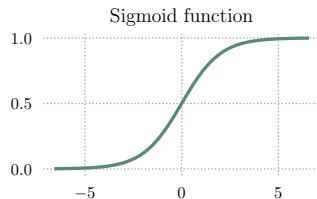
В модели **логистической регрессии** вероятность объекта  $x = (x_1, \dots, x_m)$  принадлежать классу 1 моделируется следующим образом:

$$P(y = 1|x) = \sigma(b_0 + b_1x_1 + \dots + b_mx_m),$$

где  $\sigma(z)$  — **логистическая функция** или **сигмоида**:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Сигмоида принимает в качестве аргумента вещественное число, а отдаёт число из промежутка  $[0, 1]$ .



*Замечание.* Как и ранее в линейной регрессии, мы вводим дополнительный нулевой фактор  $x_0$ , равный 1 для каждого объекта, который нужен просто чтобы записать выражение в векторном виде:

$$P(y = 1|x) = \sigma(x \cdot b),$$

где  $b = (b_0, b_1, \dots, b_m)$  — вектор коэффициентов модели.



*Замечание.* Как и ранее в линейной регрессии, мы вводим дополнительный нулевой фактор  $x_0$ , равный 1 для каждого объекта, который нужен просто чтобы записать выражение в векторном виде:

$$P(y = 1|x) = \sigma(x \cdot b),$$

где  $b = (b_0, b_1, \dots, b_m)$  — вектор коэффициентов модели.

Для оптимизации параметров модели используется **метод максимального правдоподобия**. Его схему можно изобразить следующим образом:

$$\prod_{i=1}^n P(y = y_i | x = x_i) \rightarrow \max_b$$

По сути мы подбираем набор параметров так, чтобы *максимизировать вероятность наблюдать ту выборку, которая у нас есть*.

Тут кроме формулы для  $P(y = 1|x)$  нам понадобится также формула вероятности принадлежности объекта к нулевому классу:

$$P(y = 0|x) = 1 - \sigma(x \cdot b)$$

Отсюда запишем общую вероятность:

$$P(y|x) = \sigma(x \cdot b)^y \cdot (1 - \sigma(x \cdot b))^{1-y}$$

Такие вероятности и используются в методе максимального правдоподобия.

В практическом смысле удобнее оптимизировать не саму функцию, а её логарифм (поскольку в этом случае множители превращаются в слагаемые). Кроме того, традиционно оптимизационные задачи записываются именно как задачи *минимизации*, поэтому добавим перед всем выражением минус.

Итак, минимизируется функционал:

$$Q(b) = - \sum_{i=1}^n \left[ y_i \cdot \ln(\sigma(x_i \cdot b)) + (1 - y_i) \cdot \ln(1 - \sigma(x_i \cdot b)) \right],$$

где  $x_i$  — набор признаков  $i$ -го объекта,  $y_i$  — его метка (0 или 1). Такой функционал называют *бинарной кросс-энтропией*.

Для нахождения оптимального решения используют оптимизационные методы, например, **градиентный спуск**. В нём используется вектор **градиента**, который состоит из частных производных функционала  $Q(b)$  по переменным  $b_j$ :

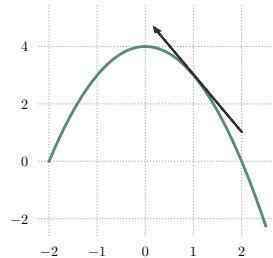
$$\nabla Q = \left( \frac{\partial Q}{\partial b_0}, \dots, \frac{\partial Q}{\partial b_m} \right)$$

Результат взятия каждой частной производной вычисляется по формуле:

$$\frac{\partial Q}{\partial b_j} = \sum_{i=1}^n (\sigma(b_0 x_{i0} + \dots + b_m x_{im}) - y_i) x_{ij},$$

где  $x_{ij}$  —  $j$ -й признак  $i$ -го объекта из выборки.

Вектор градиента указывает направление **наискорейшего роста**.



Непосредственно метод градиентного спуска заключается в следующем. Сначала выбираются начальные значения параметров  $b_0, \dots, b_m$ , т.е. вектор  $b^{[0]}$ . Затем итеративно повторяется вычисление:

$$b^{[k+1]} = b^{[k]} - \lambda_k \nabla Q(b^{[k]})$$

Иначе говоря, на каждой итерации мы делаем шаг размера  $\lambda_k$  против направления вектора градиента.

**Почему против?** Потому что вектор градиента указывает направление наискорейшего роста, следовательно, обратное направление является направлением *наискорейшего убывания*.

Описанный выше процесс повторяется, пока соседние векторы  $b^{[k+1]}$ ,  $b^{[k]}$  не перестанут сильно отличаться друг от друга.

Результат оптимизации очень сильно зависит от выбора начальных коэффициентов  $b^{[0]}$ , а также параметров  $\lambda_k$ , отвечающих за скорость спуска.

Малые значения  $\lambda_k$  приводят к низкой скорости спуска, тогда как высокие значения  $\lambda_k$  могут привести к расхождению метода.

Выбор этих значений — важная задача в теории оптимизации.

Для оценки качества моделей бинарной классификации используют **confusion matrix**, или **матрицу ошибок**:

$$M = \begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix},$$

где:


- $TP$  — *true positive*, т.е. число объектов, верно причисленных к классу 1,
- $FP$  — *false positive*, число ошибок первого рода, т.е. объектов, ложно причисленных к классу 1,
- $FN$  — *false negative*, ошибки второго рода,
- $TN$  — *true negative*, число объектов, верно причисленных к классу 0.

На основании матрицы ошибок можно получить многие известные метрики качества:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}$$





Спасибо за внимание