

# Лекция 6

## Многоклассовая классификация и категориальные признаки

Е. А. Соколов  
ФКН ВШЭ

13 октября 2016 г.

Ранее мы разобрались с общим подходом к решению задачи бинарной классификации, а также изучили свойства двух конкретных методов: логистической регрессии и метода опорных векторов. Теперь мы перейдём к более общей, многоклассовой постановке задачи классификации, и попытаемся понять, как можно свести её к уже известным нам методам.

Также мы обсудим методы работы с категориальными признаками и поймём, почему на таких данных чаще всего используются линейные модели.

## 1 Многоклассовая классификация

В данном разделе будем считать, что каждый объект относится к одному из  $K$  классов:  $\mathbb{Y} = \{1, \dots, K\}$ .

### §1.1 Сведение к серии бинарных задач

Мы уже подробно изучили задачу бинарной классификации, и поэтому вполне естественно попытаться свести многоклассовую задачу к набору бинарных. Существует достаточно много способов сделать это — мы перечислим лишь два самых популярных, а об остальных можно почитать, например, [1, раздел 4.2.7]

**Один против всех (one-versus-all).** Обучим  $K$  линейных классификаторов  $b_1(x), \dots, b_K(x)$ , выдающих оценки принадлежности классам  $1, \dots, K$  соответственно. Например, в случае с линейными моделями эти модели будут иметь вид

$$b_k(x) = \langle w_k, x \rangle + w_{0k}.$$

Классификатор с номером  $k$  будем обучать по выборке  $(x_i, 2[y_i = k] - 1)_{i=1}^\ell$ ; иными словами, мы учим классификатор отличать  $k$ -й класс от всех остальных.

Итоговый классификатор будет выдавать класс, соответствующий самому уверенному из бинарных алгоритмов:

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} b_k(x).$$

Проблема данного подхода заключается в том, что каждый из классификаторов  $b_1(x), \dots, b_K(x)$  обучается на своей выборке, и выходы этих классификаторов могут иметь разные масштабы, из-за чего сравнивать их будет неправильно [2]. Нормировать вектора весов, чтобы они выдавали ответы в одной и той же шкале, не всегда может быть разумным решением — так, в случае с SVM веса перестанут являться решением задачи, поскольку нормировка изменит норму весов.

**Все против всех (all-versus-all).** Обучим  $C_K^2$  классификаторов  $a_{ij}(x)$ ,  $i, j = 1, \dots, K$ ,  $i \neq j$ . Например, в случае с линейными моделями эти модели будут иметь вид

$$b_k(x) = \text{sign}(\langle w_k, x \rangle + w_{0k}).$$

Классификатор  $a_{ij}(x)$  будем настраивать по подвыборке  $X_{ij} \subset X$ , содержащей только объекты классов  $i$  и  $j$ :

$$X_{ij} = \{(x_n, y_n) \in X \mid [y_n = i] = 1 \text{ или } [y_n = j] = 1\}.$$

Соответственно, классификатор  $a_{ij}(x)$  будет выдавать для любого объекта либо класс  $i$ , либо класс  $j$ .

Чтобы классифицировать новый объект, подадим его на вход каждого из построенных бинарных классификаторов. Каждый из них проголосует за свой класс; в качестве ответа выберем тот класс, за который наберется больше всего голосов:

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} \sum_{i=1}^K \sum_{j \neq i} [a_{ij}(x) = k].$$

## §1.2 Многоклассовая логистическая регрессия

Некоторые методы бинарной классификации можно напрямую обобщить на случай многих классов. Выясним, как это можно проделать с логистической регрессией.

В логистической регрессии для двух классов мы строили линейную модель  $b(x) = \langle w, x \rangle + w_0$ , а затем переводили её прогноз в вероятность с помощью сигмоидной функции  $\sigma(z) = \frac{1}{1 + \exp(-z)}$ . Допустим, что мы теперь решаем многоклассовую задачу и построили  $K$  линейных моделей  $b_k(x) = \langle w_k, x \rangle + w_{0k}$ , каждая из которых даёт оценку принадлежности объекта одному из классов. Как преобразовать вектор оценок  $(b_1(x), \dots, b_K(x))$  в вероятности? Для этого можно воспользоваться оператором  $\text{SoftMax}(z_1, \dots, z_K)$ , который производит «нормировку» вектора:

$$\text{SoftMax}(z_1, \dots, z_K) = \left( \frac{\exp(z_1)}{\sum_{k=1}^K \exp(z_k)}, \dots, \frac{\exp(z_K)}{\sum_{k=1}^K \exp(z_k)} \right).$$

В этом случае вероятность  $k$ -го класса будет выражаться как

$$P(y = k \mid x, w) = \frac{\exp(\langle w_k, x \rangle + w_{0k})}{\sum_{j=1}^K \exp(\langle w_j, x \rangle + w_{0j})}.$$

Обучать эти веса предлагается с помощью метода максимального правдоподобия — так же, как и в случае с двухклассовой логистической регрессией:

$$\sum_{i=1}^{\ell} \log P(y = y_i | x_i, w) \rightarrow \max_{w_1, \dots, w_K}.$$

### §1.3 Многоклассовый метод опорных векторов

(данный материал является опциональным)

В алгоритме «один против всех» мы *независимо* строили свой классификатор за каждый класс. Попробуем теперь строить эти классификаторы одновременно, в рамках одной оптимизационной задачи. Подходов к обобщению метода опорных векторов на многоклассовый случай достаточно много; мы разберём способ, описанный в работе [3].

Для простоты будем считать, что в выборке имеется константный признак, и не будет явно указывать сдвиг  $b$ . Будем настраивать  $K$  наборов параметров  $w_1, \dots, w_K$ , и итоговый алгоритм определим как

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} \langle w_k, x \rangle.$$

Рассмотрим следующую функцию потерь:

$$\max_k \left\{ \langle w_k, x \rangle + 1 - [k = y(x)] \right\} - \langle w_{y(x)}, x \rangle. \quad (1.1)$$

Разберемся сначала с выражением, по которому берется максимум. Если  $k = y(x)$ , то оно равно  $\langle w_k, x \rangle$ ; в противном же случае оно равно  $\langle w_k, x \rangle + 1$ . Если оценка за верный класс больше оценок за остальные классы хотя бы на единицу, то максимум будет достигаться на  $k = y(x)$ ; в этом случае потеря будет равна нулю. Иначе же потеря будет больше нуля. Здесь можно увидеть некоторую аналогию с бинарным SVM: мы штрафует не только за неверный ответ на объекте, но и за неуверенную классификацию (за попадание объекта в разделяющую полосу).

Рассмотрим сначала линейно разделимую выборку — т.е. такую, что существуют веса  $w_{1*}, \dots, w_{K*}$ , при которых потеря (1.1) равна нулю. В бинарном SVM мы строили классификатор с максимальным отступом. Известно, что аналогом отступа для многоклассового случая является норма Фробениуса матрицы  $W$ ,  $k$ -я строка которой совпадает с  $w_k$ :

$$\rho = \frac{1}{\|W\|^2} = \frac{1}{\sum_{k=1}^K \sum_{j=1}^d w_{kj}^2}.$$

Получаем следующую задачу:

$$\begin{cases} \frac{1}{2} \|W\|^2 \rightarrow \min_W \\ \langle w_{y_i}, x_i \rangle + [y_i = k] - \langle w_k, x_i \rangle \geq 1, \quad i = 1, \dots, \ell; k = 1, \dots, K. \end{cases} \quad (1.2)$$

Перейдем теперь к общему случаю. Как и в бинарном методе опорных векторов, перейдем к мягкой функции потерь, введя штрафы за неверную или неуверенную

классификацию. Получим задачу

$$\begin{cases} \frac{1}{2}\|W\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{W, \xi} \\ \langle w_{y_i}, x_i \rangle + [y_i = k] - \langle w_k, x_i \rangle \geq 1 - \xi_i, \quad i = 1, \dots, \ell; k = 1, \dots, K; \\ \xi_i \geq 0, \quad i = 1, \dots, \ell. \end{cases} \quad (1.3)$$

Решать задачу (1.3) можно, например, при помощи пакета  $\text{SVM}^{\text{multiclass}}$ .

Отметим, что такой подход решает проблему с несоизмеримостью величин, выдаваемых отдельными классификаторами (о которой шла речь в подходе «один против всех»): классификаторы настраиваются одновременно, и выдаваемые ими оценки должны правильно соотноситься друг с другом, чтобы удовлетворять ограничениям.

## §1.4 Метрики качества многоклассовой классификации

В многоклассовых задачах, как правило, стараются свести подсчет качества к вычислению одной из рассмотренных выше двухклассовых метрик. Выделяют два подхода к такому сведению: микро- и макро-усреднение.

Пусть выборка состоит из  $K$  классов. Рассмотрим  $K$  двухклассовых задач, каждая из которых заключается в отделении своего класса от остальных, то есть целевые значения для  $k$ -й задачи вычисляются как  $y_i^k = [y_i = k]$ . Для каждой из них можно вычислить различные характеристики (ТР, ФР, и т.д.) алгоритма  $a^k(x) = [a(x) = k]$ ; будем обозначать эти величины как  $\text{TP}_k, \text{FP}_k, \text{FN}_k, \text{TN}_k$ . Заметим, что в двухклассовом случае все метрики качества, которые мы изучали, выражались через эти элементы матрицы ошибок.

При микро-усреднении сначала эти характеристики усредняются по всем классам, а затем вычисляется итоговая двухклассовая метрика — например, точность, полнота или F-мера. Например, точность будет вычисляться по формуле

$$\text{precision}(a, X) = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FP}}},$$

где, например,  $\overline{\text{TP}}$  вычисляется по формуле

$$\overline{\text{TP}} = \frac{1}{K} \sum_{k=1}^K \text{TP}_k.$$

При макро-усреднении сначала вычисляется итоговая метрика для каждого класса, а затем результаты усредняются по всем классам. Например, точность будет вычислена как

$$\text{precision}(a, X) = \frac{1}{K} \sum_{k=1}^K \text{precision}_k(a, X); \quad \text{precision}_k(a, X) = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}.$$

Если какой-то класс имеет очень маленькую мощность, то при микро-усреднении он практически никак не будет влиять на результат, поскольку его вклад в средние ТР, ФР, FN и TN будет незначителен. В случае же с макро-вариантом усреднение проводится для величин, которые уже не чувствительны к соотношению размеров классов (если мы используем, например, точность или полноту), и поэтому каждый класс внесет равный вклад в итоговую метрику.

## 2 Классификация с пересекающимися классами

Усложним постановку задачи. Будем считать, что в задаче  $K$  классов, но теперь они могут *пересекаться* — каждый объект может относиться одновременно к нескольким классам. Это означает, что каждому объекту  $x$  соответствует вектор  $y \in \{0, 1\}^K$ , показывающий, к каким классам данный объект относится. Соответственно, обучающей выборке  $X$  будет соответствовать матрица  $Y \in \{0, 1\}^{\ell \times K}$ , описывающая метки объектов; её элемент  $y_{ik}$  показывает, относится ли объект  $x_i$  к классу  $k$ . Данная задача в англоязычной литературе носит название *multi-class classification*. К ней может относиться, например, определение тэгов для фильма или категорий для статьи на Википедии.

### §2.1 Независимая классификация (Binary relevance)

Самый простой подход к решению данной задачи — предположить, что все классы независимы, и определять принадлежность объекта к каждому отдельным классификатором. Это означает, что мы обучаем  $K$  бинарных классификаторов  $a_1(x), \dots, a_K(x)$ , причём классификатор  $b_k(x)$  обучается по выборке  $(x_i, y_{ik})_{i=1}^{\ell}$ . Для нового объекта  $x$  целевая переменная оценивается как  $(a_1(x), \dots, a_K(x))$ .

Основная проблема данного подхода состоит в том, что никак не учитываются возможные связи между отдельными классами. Тем не менее, такие связи могут иметь место — например, категории на Википедии имеют древовидную структуру, и если мы с большой уверенностью отнесли статью к некоторой категории, то из этого может следовать, что статья относится к одной из категорий-потомков.

### §2.2 Стекинг классификаторов

Для учёта корреляций между классами можно воспользоваться следующим несложным подходом. Разобьём обучающую выборку  $X$  на две части  $X_1$  и  $X_2$ . На первой части обучим  $K$  независимых классификаторов  $b_1(x), \dots, b_K(x)$ . Далее сформируем для каждого объекта  $x_i \in X_2$  из второй выборки признаковое описание, состоящее из прогнозов наших классификаторов:

$$x'_{ik} = b_k(x_i), \quad x_i \in X_2,$$

получив тем самым выборку  $X'_2$ . Обучим на ней новый набор классификаторов  $a_1(x), \dots, a_K(x)$ , каждый из которых определяет принадлежность объекта к одному из классов. При этом все новые классификаторы опираются на прогнозы классификаторов первого этапа  $b_1(x), \dots, b_K(x)$ , и поэтому могут обнаружить связи между различными классами. Такой подход называется *стекингом* и достаточно часто используется в машинном обучении для усиления моделей.

Отметим, что обучать классификаторы  $b_k(x)$  и  $a_k(x)$  на одной и той же выборке было бы плохой идеей. Прогнозы базовых моделей  $b_k(x)$  содержат в себе информацию об обучающей выборке  $X_1$ ; получается, что новые признаки  $x'_{ik} = b_k(x_i)$ , посчитанные по этой же выборке, по сути будут «подглядывать» в целевую переменную, и обучение на них новой модели просто приведёт к переобучению.

Посмотрим на эту проблему несколько иначе. Допустим, мы обучили на выборке  $X_1$  алгоритм  $b(x)$ , а затем на этой же выборке обучили второй алгоритм  $a(b(x))$ ,

использующий в качестве единственного признака результат работы  $b(x)$ . Если модель  $b(x)$  не переобучилась и будет показывать на новых данных такое же качество, как и на обучающей выборке, то никаких проблем не будет. Тем не менее, обычно модели хотя бы немного переобучаются. Будем считать, что на обучении  $b(x)$  имеет среднее отклонение от целевой переменной в 5%, а на новых данных она в среднем ошибается на 10% из-за переобучения. Тогда модель  $a(x)$  будет рассчитывать на среднее отклонение в 5%, но на новых данных ситуация будет другой — фактически, изменится распределение её признака, что приведёт к не самым лучшим последствиям.

## §2.3 Трансформация пространства ответов

Существуют подходы, которые пытаются в рамках одной модели учитывать взаимосвязи между классами. Один из них [4] предлагает преобразовать пространство ответов так, что классы оказались как можно менее зависимыми. Это можно сделать с помощью сингулярного разложения матрицы  $Y$ :

$$Y = U\Sigma V^T.$$

Известно, что если в этом разложении занулить все диагональные элементы матрицы  $\Sigma$  кроме  $m$  наибольших, то мы получим матрицу, наиболее близкую к  $Y$  с точки зрения нормы Фробениуса среди всех матриц ранга  $m$ .

Обозначим через  $V_M$  матрицу, состоящую из тех  $M$  столбцов матрицы  $V$ , которые соответствуют наибольшему сингулярным числам. Спроецируем с её помощью матрицу  $Y$ :

$$Y' = YV_M \in \mathbb{R}^{\ell \times M}.$$

Поскольку столбцы матрицы  $V_M$  ортогональны, то можно рассчитывать, что после проекции на них метки станут менее зависимыми. Настроим на новые метки  $Y'$   $M$  независимых моделей  $a_1(x), \dots, a_M(x)$ . Обозначим матрицу прогнозов для нашей выборки через  $A' \in \mathbb{R}^{\ell \times M}$ . Чтобы получить оценки принадлежности исходным классам, переведём матрицу  $A'$  в исходное пространство:

$$A = A'V_M^T.$$

Далее в лекциях мы будем изучать метод главных компонент и увидим, что описанный подход, по сути, аналогичен применению данного метода к матрице меток.

## §2.4 Метрики качества классификации с пересекающимися классами

Обозначим через  $Y_i$  множество классов, которым объект  $x_i$  принадлежит на самом деле, а через  $Z_i$  — множество классов, к которым объект был отнесён алгоритмом  $a(x)$ .

Вполне логичной мерой ошибки будет хэммингово расстояние между этими множествами — то есть доля классов, факт принадлежности которым угадан неверно:

$$\text{hamming}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \setminus Z_i| + |Z_i \setminus Y_i|}{K}.$$

Данную метрику необходимо минимизировать.

Стандартные метрики качества классификации можно обобщить на multilabel-задачу так же, как и на случай с непересекающимися классами — через микро- или макро-усреднение. Есть и несколько другой подход к обобщению основных метрик качества:

$$\text{accuracy}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|},$$

$$\text{precision}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Z_i|},$$

$$\text{recall}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Y_i|}.$$

Все эти метрики необходимо максимизировать.

## 3 Категориальные признаки

Допустим, в выборке имеется категориальный признак, значение которого на объекте  $x$  будем обозначать через  $f(x)$ . Будем считать, что он принимает значения из множества  $U = \{u_1, \dots, u_n\}$ . Чтобы использовать такой признак в линейных моделях, необходимо сначала его закодировать. Существует много подходов к использованию категориальных признаков — о многих из них можно узнать в работе [5], мы же рассмотрим несколько наиболее популярных.

### §3.1 Бинарное кодирование (one-hot encoding)

Простейший способ — создать  $n$  индикаторов, каждый из которых будет отвечать за одно из возможных значений признака. Иными словами, мы формируем  $n$  бинарных признаков  $g_1(x), \dots, g_n(x)$ , которые определяются как

$$g_j(x) = [f(x) = u_j].$$

Главная проблема этого подхода заключается в том, что на выборках, где категориальные признаки имеют миллионы возможных значений, мы получим огромное количество признаков. Линейные модели хорошо справляются с такими ситуациями за счёт небольшого количества параметров и достаточно простых методов обучения, и поэтому их часто используют на выборках с категориальными признаками.

### §3.2 Бинарное кодирование с хэшированием

Рассмотрим модификацию бинарного кодирования, которая позволяет ускорить процесс вычисления признаков. Выберем хэш-функцию  $h : U \rightarrow \{1, 2, \dots, B\}$ , которая переводит значения категориального признака в числа от 1 до  $B$ . После этого бинарные признаки можно индексировать значениями хэш-функции:

$$g_j(x) = [h(f(x)) = j], \quad j = 1, \dots, B.$$



Основное преимущество этого подхода состоит в том, что отпадает необходимость в хранении соответствий между значениями категориального признака и индексами бинарных признаков. Теперь достаточно лишь уметь вычислять саму хэш-функцию, которая уже автоматически даёт правильную индексацию.

Также хэширование позволяет понизить количество признаков (если  $B < |U|$ ), причём, как правило, это не приводит к существенной потере качества. Это можно объяснить и с помощью интуиции — если у категориального признака много значений, то, скорее всего, большая часть этих значений крайне редко встречается в выборке, и поэтому не несёт в себе много информации; основную ценность представляют значения  $u \in U$ , которые много раз встречаются в выборке, поскольку для них можно установить связи с целевой переменной. Хэширование, по сути, случайно группирует значения признака — в одну группу попадают значения, получающие одинаковые индексы  $h(u)$ . Поскольку «частых» значений не так много, вероятность их попадания в одну группу будет ниже, чем вероятность группировки редких значений.

### §3.3 Счётчики

Попробуем закодировать признаки более экономно. Заметим, что значения категориального признака нужны нам не сами по себе, а лишь для предсказания класса. Соответственно, если два возможных значения  $u_i$  и  $u_j$  характерны для одного и того же класса, то можно их и не различать.

Определим наш способ кодирования. Вычислим для каждого значения  $u$  категориального признака  $(K + 1)$  величин:

$$\begin{aligned} \text{counts}(u, X) &= \sum_{(x,y) \in X} [f(x) = u], \\ \text{successes}_k(u, X) &= \sum_{(x,y) \in X} [f(x) = u][y = k], \quad k = 1, \dots, K. \end{aligned}$$

По сути, мы посчитали количество объектов с данным значением признака, а также количество объектов различных классов среди них.

После того, как данные величины подсчитаны, заменим наш категориальный признак  $f(x)$  на  $K$  вещественных  $g_1(x), \dots, g_K(x)$ :

$$g_k(x, X) = \frac{\text{successes}_k(f(x), X) + c_k}{\text{counts}(f(x), X) + \sum_{m=1}^K c_m}, \quad k = 1, \dots, K.$$

Здесь признак  $g_k(x)$  фактически оценивает вероятность  $p(y = k | f(x))$ . Величины  $c_k$  являются своего рода регуляризаторами и предотвращают деление на ноль в случае, если не найдётся объектов одного из классов. Для простоты можно полагать их все равными единице:  $c_1 = \dots = c_K = 1$ . У признаков  $g_k(x)$  есть много названий: счётчики<sup>1</sup>, правдоподобия и т.д.

Отметим, что  $g_k(x)$  можно воспринимать как простейший классификатор — значит, при обучении полноценного классификатора на признаках-счётчиках мы

---

<sup>1</sup><http://blogs.technet.com/b/machinelearning/archive/2015/02/17/big-learning-made-easy-with-counts.aspx>



рискуем столкнуться с переобучением из-за «утечки» целевой переменной в значения признаков (мы уже обсуждали эту проблему, разбираясь со стекингом). Чтобы избежать переобучения, как правило, пользуются подходом, аналогичным кросс-валидации. Выборка разбивается на  $m$  частей  $X_1, \dots, X_m$ , и для подвыборки  $X_i$  значения признаков вычисляются на основе статистик, подсчитанных по всем остальным частям:

$$x \in X_i \Rightarrow g_k(x) = g_k(x, X \setminus X_i).$$

Можно взять число блоков разбиения равным числу объектов  $m = \ell$  — в этом случае значения признаков для каждого объекта будут вычисляться по статистикам, подсчитанным по всем остальным объектам.

При использовании счётчиков нередко используют следующие трюки:

1. К признакам можно добавлять не только дроби  $g_k(x)$ , но и значения  $\text{counts}(f(x), X)$  и  $\text{successes}_k(f(x), X)$ .
2. Можно сгенерировать парные категориальные признаки, т.е. для каждой пары категориальных признаков  $f_i(x)$  и  $f_j(x)$  создать новый признак  $f_{ij}(x) = (f_i(x), f_j(x))$ . После этого счётчики можно вычислить и для парных признаков; при этом общее количество признаков существенно увеличится, но при этом, как правило, прирост качества тоже оказывается существенным.
3. Если у категориальных признаков много возможных значений, то хранение статистик  $\text{counts}(u, X)$  и  $\text{successes}_k(u, X)$  может потребовать существенного количества памяти. Для экономии памяти можно хранить статистику не по самим значениям категориального признака  $u \in U$ , а по хэшам от этих значений  $h(u)$ . Регулируя количество возможных значений хэш-функции, можно ограничивать количество используемой памяти.
4. Можно вычислять несколько счётчиков для разных значений параметров  $c_1, \dots, c_K$ .
5. Можно все редкие значения категориального признака объединить в одно, поскольку скорее всего, для редких значений не получится качественно оценить статистики  $\text{successes}_k$  и  $\text{counts}$ . Благодаря этому можно будет сократить расходы на память при хранении статистики. Более того, можно предположить, что все редкие значения похожи, и относить к данной «объединённой» группе и новые значения признака, которые впервые встретятся на тестовой выборке.

Отметим, что данный подход работает только для задач классификации. В то же время можно попытаться адаптировать его и для задач регрессии, вычисляя несколько бинаризаций целевой переменной по разным порогам, и для каждой такой бинаризации вычисляя счётчики.

## Список литературы

- [1] Мерков. А. Б. Введение в методы статистического обучения. // <http://www.recognition.mccme.ru/pub/RecognitionLab.html/slbook.pdf>

- [2] *Bishop, C.M.* Pattern Recognition and Machine Learning. // Springer, 2006.
- [3] *Crammer, K., Singer, Y.* On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. // Journal of Machine Learning Research, 2:265-292, 2001.
- [4] *Tai, Farbound and Lin, Hsuan-Tien.* Multilabel Classification with Principal Label Space Transformation. // Neural Comput., 24-9, 2012.
- [5] *Дьяконов А. Г.* Методы решения задач классификации с категориальными признаками. // Прикладная математика и информатика. Труды факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова. 2014.