Московский государственный университет
имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Ожерельев Илья Сергеевич

Решение многоклассовых задач распознавания

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., профессор
В.В. Рязанов

Москва, 2015

Содержание

Аннотация

В данной работе приведен алгоритм настройки параметров кодовой матрицы для методов многоклассовой классификации с использованием самокорректирующихся кодов (ECOC Error Correcting Output Codes)

Приведенный алгоритм позволяет учитывать сложность конкретной бинарной задачи при построении композиции, эффективно определять оптимальный размер кодовой матрицы, не делая дополнительных предположений о данных.

В ходе выполнения работы были проведены эксперименты на реальных данных, сравнивающие предложенный метод с классическими методами ECOC.

Предложен алгоритм, который

1. настраивает по данным размер композиции
2. поощряет включение в матрицу ECOC бинарных классификаторов, имеющих лучшее качество классификации
3. максимизирует расстояние между строками кодовой матрицы
4. учитывает качество классификации объекта бинарными классификаторами на шаге декодирования

1 Введение

1.1 Основные понятия

В работе будут рассмотрены методы решения задачи классификации с большим числом классов[12]. Напомним ее формулировку.

Пусть задано множество объектов χ, множество номеров классов $\Upsilon = \{0, \ldots, K-1\}$, и существует целевая функция $y^* : \chi \to \Upsilon$, значения которой известны на конечном множестве объектов $x_1, \ldots x_n \in \chi$. Пары $(x_i, y_i = y^*(x_i))$ называют прецедентами, а их совокупность $(x_i, y_i)_{i=1}^n$ - обучающей выборкой.

Задача обучения по прецедентам заключается в том, чтобы по обучающей выборке восстановить зависимость y^*, построив решающую функцию $\chi \to \Upsilon$, которая бы приближала целевую функцию $y^*(x)$, не только на объектах обучающей выборки, но и на всем множестве χ . Решающая функция должна допускать эффективную компьютерную реализацию.

Каждый объект x_i задается некоторым набором характеристик — признаков. Допустим, признаков m штук, признаки - вещественные числа. Тогда каждому объекту $x_i \in \chi$ соответствует вектор-строка $(x_i^1, \ldots, x_i^m)$ — признаковое описание. Таким образом обучающую выборку можно представить в виде матрицы $X \in \mathbb{R}^{n \times m}$.

В данной работе будут рассмотрены некоторые задачи теории кодирования.

Пусть нам надо передать слово u (двоичный вектор длины L) по зашумленному каналу. Из-за шума возможны ошибки типа "замещение". То есть с некоторой

вероятностью q значение бита в передаваемом сообщении инвертируется.

Для этого слово u преобразуется в сообщение v (двоичный вектор длины $N > L$). Между сообщением и словом существует взаимно однозначное соответствие. После передачи по зашумленному каналу, v преобразуется в вектор w (двоичный вектор длины N). Задача состоит в том, чтобы по вектору w восстановить сообщение v и слово u.

$u^{1\times L} \to$ {шифрование } $\to v^{1\times N} \to$ {шум } $\to w^{1\times N} \to$ {дешифровка} $\to v^{1\times N} \to u^{1\times L}$

Кодовой матрицей называют двоичную матрицу, строками которой являются сообщения. Кодовая матрица имеет размер $W \times N$, где W - количество слов.

Коды, исправляющие ошибки, используются для решения задач многоклассовой классификации[6]. Самокорректирующиеся коды позволяют применить композицию бинарных классификаторов для решения задачи многоклассовой классификации. Данный метод показал свою эффективность для решения множества прикладных задач: распознавание лиц[1], распознавание шрифтов[2], опознание человека по фотографии[3].

Идею метода проиллюстрируем на примере. Пусть даны признаковые описания объектов $X \in \mathbb{R}^{n\times m}$, их метки $y \in \{0, \ldots, 5\}^{n\times 1}$. Требуется разделить объекты на 6 классов.
Закодируем номера классов двоичными векторами

	1	2	3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1

Получим матрицу размера 6×3. Эта матрица называется кодовой. Рассмотрим столбцы кодовой матрицы.

Каждый столбец задает новую задачу бинарной классификации на выборке X. Если i-й элемент столбца равен 0, то в соответствующей бинарной задаче все объекты i-ого класса имеют метку 0, если элемент столбца равен 1, все объекты соответствующего класса имеют метку 1.

Первый столбец приведенной выше матрицы присваивает объектам классов 1 - 4 метку 0, объектам классов 5 - 6 присваивает метку 1.

Таким образом, каждый столбец соответствует задаче бинарной классификации для выборки X и новых меток объектов $y^j \in \{0, 1\}^{n\times 1}$, где j - номер столбца кодовой матрицы.

Каждая строка кодовой матрицы - двоичный вектор, кодирующий номер класса. Для вновь поступившего объекта x составим вектор ответов бинарных классификаторов $(b^1(x), b^2(x), b^3(x))$ и определим по нему класс объекта, сравнив вектор ответов со строками кодовой матрицы.

Методы, с использованием ECOC, основаны на построении композиции классификаторов. Кодовая матрица задает композицию классификаторов. Вид кодовой матрицы является одним из параметров метода. В приведенном выше примере подбор кодовой матрицы не проводился.

Отметим недостатки рассмотренной матрицы:

1. Ошибка одного из классификаторов приводит к ошибке композиции.
2. Различные задачи классификации имеют различную сложность. Не показано, что входящие в матрицу столбцы соответствуют достаточно простым задачам.

Для кодирования номеров классов можно применить самокорректирующийся код. Шум канала - ошибки классификации.

Опишем метод ECOC формально. Метод решает задачу классификации на K классов. Метод ECOC, может быть разделен на две подзадачи:

1. кодирование
2. декодирование

На этапе кодирования каждому классу ставится в однозначное соответствие двоичное кодовое слово(вектор - строка $m_j \in \{0,1\}^{1\times N}$). Также на первом этапе определяется правило, по которому каждому распознаваемому объекту присваивается некоторое значение (вектор - строка $A_i \in \{0,1\}^{1\times N}$).

Все слова $m_j, j \in 1,\ldots,K$ имеют одинаковую длину, таким образом из них можно составить кодовую матрицу $M \in \{0,1\}^{K\times N}$. Слова - строки матрицы. Каждый столбец кодовой матрицы - представляет собой задачу для бинарного классификатора. Объекты, для которых j-й элемент кодового слова m равен 0, объединяются в первый "макрокласс"j-й задачи. Объекты, для которых j-й элемент кодового слова равен 1, объединяются во второй "макрокласс"j-й задачи.

На этапе декодирования для задач бинарной классификации, соответствующих столбцам матрицы, обучаются классификаторы $b^1, b^2, \ldots, b^N$ То, каким образом строится кодовая матрица и правило выбора ближайшего класса определяют метод ECOC. При этом, можно определить минимальное расстояние между строками кодовой матрицы. Если это расстояние равно D, то при определении ближайшего класса мы можем исправить $\frac{D-1}{2}$ ошибки, если мы определяем ближайшее слово расстоянием Хемминга.

Заметим, что метод ECOC состоит в использовании одного мета - алгоритма (алгоритма декодирования) над результатами работы других базовых алгоритмов (бинарных классификаторов).

При построении композиций алгоритмов рекомендуется подбирать оптимальные алгоритмы и длину композиции.[11,12]

Кодовую матрицу, минимизирующую чисто ошибок, совершаемых на тестовой выборке или максимизирующую другой, заданный в задаче функционал качества классификации, будем называть оптимальной.

1.2 Использование метода ECOC в моделях KNN, SVM и Байесовских методах

Прежде чем подробнее рассмотреть задачу построения кодовой матрицы, обсудим актуальность задачи.

Многие методы машинного обучения, например KNN, SVM, Байесовские методы, легко обобщаются для решения многоклассовых задач. В этом разделе будет показано, что эти обобщения можно переформулировать в терминах методов ECOC и свести их к важному частному случаю - методу "один против всех".

Метод "один против всех"является частным случаем (методом с диагональной кодовой матрицей и несколько более сложной процедурой декодирования). Рассмотрим процедуру декодирования.

№ класса/классификатор	b^1	b^2	$\dots$	b^N
1	1	0	$\dots$	0
2	0	1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
K	0	0	$\dots$	1

Любой двухклассовый классификатор дает оценку принадлежности объекта к первому классу и оценки принадлежности объекта ко второму классу [12]. Алгоритм относит объект к тому классу, оценка за который больше.

$$a_j^i = b^i(x_j) = argmax_{a\in\{0,1\}}(\mathbb{I}(a=1) * est_1(x_j) + \mathbb{I}(a=0) * est_2(x_j)), i \in \{1,\dots,N\}$$

где $\mathbb{I}$ - индикатор, $est_k(x_j)$ - оценка объекта x_j за класс k.

Для каждой бинарной задачи введем следующие обозначения: объекты класса исходной задачи, который помечен единицей, отнесем к первому классу соответствующей бинарной задачи, а остальные объекты ко второму.

Построим вектор оценок классификаторов за первый класс.

$$Est = (est_1^1, est_1^2, \dots, est_1^N)$$

Вектор ответов классификаторов: $A \in \{0,1\}^K$.

$$A_i = 1 <=> Est_i = \max_i(Est)$$

Такая процедура позволяет лучше обрабатывать ситуации, в которых несколько бинарных классификаторов допустили ошибку. То есть, два или более классификаторов отнесли объект к первому классу, или же все K алгоритмов отнесли объект ко второму классу.

Если несколько классификаторов относят объект к первому классу, в окончательном ответе композиции будет учтен только классификатор с наибольшим отступом от разделяющей поверхности.
Заметим, что метод "одни против всех"обладает рядом недостатков. Например, не доказана оптимальность кодовой матрицы.

Рассмотрим обобщенный алгоритм декодирования.

$$Est^* = (\frac{1}{1+e^{est_2^1-est_1^1}}, \frac{1}{1+e^{est_2^2-est_1^2}}, \ldots, \frac{1}{1+e^{est_2^N-est_1^N}})$$

Ответом композиции является номер строки кодовой матрицы, ближайшей к Est^* по евклидовому расстоянию.

Заметим, что в случае единичной кодовой матрицы, рассматриваемая функция декодирования будет давать тот же результат, что и метод "один против всех". Кроме того, чем больше расстояние Хемминга между строками матрицы, тем качественнее будет происходить декодирование.

Отметим, что расстояние между строками матрицы и ответом композиции может быть произвольной метрикой. Для сглаживания оценок не обязательно применять сигмоиду. Главное, чтобы сглаживающая функция преобразовывала оценки в отрезок $[0, 1]$, так, что объекты с маленьким отступом попадали в район 0.5

KNN

Подробно метод KNN изложен в [12]. Основная идея состоит в том, что в пространстве объектов вводится расстояние. Далее для распознаваемого объекта находятся K ближайших из тестовой выборки. Объекту присваивается тот класс, который имеют большинство соседей.

Заметим, что среди оценок за класс выбирается максимум. Так работает "один против всех".

В качестве оценки за i-й класс можно взять количество объектов i-ого класса среди соседей.

Байесовские методы

Подробнее о вероятностной постановке задачи классификации можно прочитать в [12].

X — множество объектов, Y — конечное множество имён классов, множество $X \times Y$ является вероятностным пространством с плотностью распределения $p(x, y) = P(y)p(x|y)$. Вероятности появления объектов каждого из классов $P_y = P(y)$ называются априорными вероятностями классов. Плотности распределения $p_y(x) = p(x|y)$ называются функциями правдоподобия классов.

Класс объекта определяется следующим образом:

$$\tilde{y_i} = argmax_{a \in Y} P(a)p(x_i|a)$$

$P(i)p(x_j|i)$ - оценка за i-й класс. Далее переформулировать задачу в терминах ECOC не представляет труда.

SVM

С методом опорных векторов можно ознакомиться в [12]. Рассмотрим обобщение метода SVM на многоклассовый случай.

Имеется K классов, к одному из которых надо отнести объект. Индексы классов объектов обучающей выборки будем задавать матрицей $T = (\tilde{t_1}, \ldots, \tilde{t_n})$, где $\tilde{t_i} \in \{\{0,1\}^K | \sum_{j=1}^K t_{ij} = 1\} = \tau$; $\tilde{t_i}$ — строка матрицы T, бинарный вектор, в котором присутствует лишь одна единица в той позиции, номер которой является индексом класса i-ого объекта.

В процессе обучения необходимо настроить параметры матрицы

$$W = (\tilde{w_1}, \ldots, \tilde{w_K}) \in \mathbb{R}^{K \times m}, w_j \in \mathbb{R}^{1 \times m}$$

представляющие собой наборы коэффициентов линейной комбинации признаков для каждого класса.

Решающее правило будет иметь вид $t^*(x) \in \tau, t_j^* = 1 <=> j = argmax_j(\tilde{w_j} x^T)$

Настройка осуществляется исходя из требования минимизации ошибки на обучающей выборке и минимизации нормы весов:

$$\frac{1}{2}||W||^2 + C\sum_{i=1}^{n} \xi_i \to \min_{W,\xi}$$

С ограничениями:

$$\tilde{t_i} W^T x_i^T \geq \tilde{t} W^T x_i^T + \Delta(\tilde{t}, \tilde{t_i}) - \xi_i, \forall i = 1, \ldots, n; \quad \forall \tilde{t} \in \tau$$

$$\xi_i \geq 0$$

Так как

$$argmin_W ||W|| = argmin_W \sum_{j=1}^{K} ||\tilde{w_j}||,$$

многоклассовый метод опорных векторов основан на методе "один против всех".

Заметим, что множество τ задает кодовую матрицу. Векторы из τ - строки кодовой матрицы. поэтому, в τ можно включить любые Kстрок кодовой матрицы. Ограничения:

$$(\tilde{t_i} - \alpha) W^T x_i^T \geq (\tilde{t} - \alpha) W^T x_i^T + \Delta(\tilde{t}, \tilde{t_i}) - \xi_i, \forall i = 1, \ldots, n; \ \forall \tilde{t} \in \tau; \ \alpha = (0.5, \ldots, 0.5) \in \mathbb{R}^{1 \times N}$$

$$\xi_i \geq 0$$

Для того, чтобы работать с множеством τ общего вида придется заменить решающее правило и использовать для определения класса объекта обобщенную процедуру декодирования.

Такой алгоритм имеет существенный недостаток. Добавление столбца в кодовую матрицу полностью меняет ограничения, накладываемые на матрицу W. Все $m * (N + 1)$ элементов матрицы W придется настраивать заново.

В работе применялся другой алгоритм. По кодовой матрице строилась композиция SVM. Каждый двоичный классификатор обучался независимо от остальных. Такой подход позволяет значительно сократить вычислительные затраты при изменении кодовой матрицы. При добавлении столбца нужно настроить всего m параметров.

Таким образом, большинство известных методов многоклассовой классификации основаны на методе ECOC. Метод "один против всех"позволяет неплохо обрабатывать ошибки, требует небольших вычислительных затрат. С другой стороны, метод ECOC является методом построения композиции алгоритмов. Алгоритм построения кодовой матрицы является алгоритмом построения композиции. Многие известные методы построения композиций настраивают композицию по обучающим данным. В методе "один против всех"композиция не настраивается вообще. Вопрос создания эффективного метода построения кодовой матрицы остается открытым.

1.3 Постановка задачи

В работе решается задача многоклассовой классификации при помощи метода ECOC.
$K \in \{3, \ldots, 12\}$ - число классов
$X_{tr} \in \mathbb{R}^{N_1 xK}, Y_{tr}$ - обучающая выборка
$X_{tst} \in \mathbb{R}^{N_2 xK}, Y_{tst}$ - тестовая выборка

Необходимо создать алгоритм построения кодовой матрицы:

1. по данным настраивать размер композиции
2. учитывать сложность конкретной бинарной задачи при генерации кодовой матрицы
3. учитывать качество обученных классификаторов на шаге декодирования
4. алгоритм должен допускать эффективную компьютерную реализацию.

1.4 Обзор методов построения кодовой матрицы

Разработано множество алгоритмов кодирования и декодирования для задачи передачи данных по зашумленному каналу. Например коды Боуза — Чоудхури — Хоквингема (БЧХ-коды) [9] или низкоплотностные (LDPC) коды [10]. Однако, эти результаты мало применимы для построения матрицы ECOC.

Во-первых, модель шума, на которую опирается теория кодирования предполагает, что инвертирование символа кодового слова происходит независимо с некоторой вероятностью q. В нашем случае это не так. Каждый столбец кодовой матрицы соответствует бинарному классификатору. Так как все классификаторы работают с данными из одной генеральной совокупности, ошибки классификаторов нельзя считать независимыми. Например, ошибки в повторяющихся столбцах будут происходить синхронно.

Во-вторых, в теории кодирования на множество кодовых слов зачастую накладываются существенные ограничения (линейные, циклические коды). Это связано с тем, что при большом количестве кодовых слов задача декодирования становится слишком трудной. В задах классификации это предположение

избыточно. Так как номер класса часто можно закодировать сообщением длинной в один байт, нет никаких трудностей в том, чтобы декодировать сообщение, сравнив его со всеми кодовыми словами.

Классические результаты теории кодирования не применимы для построения композиции классификаторов. Рассматриваемая задача отличается от задачи передачи сообщений по зашумленному каналу.

В статье [6] авторы метода ECOC предложили несколько подходов для построения кодовой матрицы. Во всех этих подходах максимизируется расстояние между строками кодовой матрицы, также авторы замечают, что в кодовой матрице не должно быть одинаковых столбцов (с точностью до суммы по модулю два), они соответствуют одному и тому же разбиению. Также в матрице не должно быть нулевого и единичного столбцов, они не разбивают множество объектов на два класса.

Случайная разреженная матрица. Случайным образом генерируются кодовые матрицы с различными столбцами. Из них выбирается та, которая исправит больше ошибок.

Условная дискретная оптимизация. Задача поиска оптимальной кодовой матрицы NP-полная. Поэтому в этот раздел входят все стохастические, жадные алгоритмы, которые решают эту задачу приближенно. Преимущества стохастических методов:

- Все стохастические методы имеют эффективную реализацию (время работы стохастических методов значительно меньше, чем время настройки одного бинарного классификатора)
- Легко вводить новые ограничения на столбцы матрицы

Недостатки стохастических методов

- Не гарантируется нахождение оптимального решения.
- Максимизация расстояния между строками матрицы не гарантирует нахождения оптимальной по числу ошибок классификации кодовой матрицы.

2 Алгоритм настройки параметров кодовой матрицы

2.1 Описание алгоритма

Предлагается разделить обучающую выборку на две части. На первой части обучающей выборки настроить M_0 различных бинарных классификаторов и оценить их качество. Из них отобрать $M < M_0$ лучших. Далее стохастическим методом построить из отобранных M классификаторов кодовую матрицу размера $K \times N, N < M$, оптимальную по межстрочному расстоянию.

Значения M и N предлагается настроить на второй части выборки. Для этого

необходимо вычислить ответы всех обученных классификаторов на второй части выборки и сохранить их. Таким образом, для тестирования новой матрицы не нужно будет заново вычислять ответы бинарных классификаторов. Достаточно будет только пересчитать сообщения и расстояния до строк матрицы. Эти процедуры не займут много времени.

Входные параметры: X_1, X_2 - объекты двух частей обучающей выборки; y_1 y_2 - соответствующие метки объектов; θ_{bin} - сетка значений гиперпараметров бинарных классификаторов, настраиваемых на скользящем контроле. M_0 - количество обучаемых бинарных классификаторов. θ_{ECOC} - сетка значений M и N, настраиваемая на второй части выборки, max_iter - число итераций, затрачиваемое на генерацию кодовой матрицы, num_start - число кодовых матриц, сгенерированных при данных θ_{ECOC}.

Выходные данные: кодовая матрица res_matrix и соответствующие бинарные классификаторы res_class.

Вспомогательные функции:

- $matrix = gen\_matrix(columns, max\_iter, N)$ - генерирует кодовую матрицу из N столбцов; столбцы матрицы являются элементами списка $columns$.
- $[best\_matrix, best\_qual] =$
  $test\_ecoc(columns, answers, max\_iter, N, num\_start, y_2)$.
  $answers$ - ответы классификаторов, соответствующим задачам $columns$, на выборке (X_2, y_2). Функция num_start раз вызывает $gen\_matrix(columns, max\_iter, N)$ и выбирает из полученных матриц лучшую на тестовой выборке (X_2, y_2). Функция возвращает лучшую матрицу и оценку ее качества.

Алгоритм:

1. Случайным образом выбрать M_0 различных бинарных задач.
2. Обучить M_0 соответствующих классификаторов на выборке X_1, y_1 методом 5-fold CV, с гиперпараметрами θ_{bin}.
3. Сохранить классификаторы (bin_class) и оценки их качества на скользящем контроле (bin_qual)
4. Получить ответы обученных классификаторов на X_2, y_2.
5. Сохранить ответы в список $answers$.
6. Обнулить уровень качества лучшей матрицы $res\_qual = 0$
7. Для каждой пары $(M, N) \in \theta_{ECOC}, M > N$
   (a) выбрать из bin_class M лучших классификаторов. Сохранить их ответы ($best\_answers$) и соответствующие столбцы ($best\_colums$)
   (b) вызвать $[best\_matrix, best\_qual] =$
   $test\_ecoc(best\_columns, best\_answers, max\_iter, N, num\_start, y_2)$
   (c) Если $best\_qual > res\_qual$, присвоить
   $res\_qual = best\_qual$; $res\_matrix = best\_matrix$
8. в переменную res_class записать соответствующие классификаторы

Замечание 1. Для реализации gen_matrix можно просто написать генератор случайных подмножеств $columns$ мощности N.

Замечание 2. При реализации функции $test\_ecoc$ можно сначала нормировать матрицу $answers$, а затем рассчитать расстояния от элементов $answers$ до нуля и до единицы и сохранить их в памяти. Полученный трехмерный массив подать на вход функции $test\_ecoc$. Таким образом, для того, чтобы найти расстояние от объекта до строк матрицы, нам придется всего лишь сложить заранее посчитанные значения.

Замечание 3. Функция gen_matrix не учитывает сложность входящих в нее столбцов. Рекомендуется выбирать num_start большим единицы, так как влияние качества входящих в композицию классификаторов существенно.

Предложенный алгоритм удовлетворяет поставленным выше требованиям.

1. размер композиции настраивается по данным
2. уменьшение M, поощряет включение в матрицу ECOC бинарных классификаторов, имеющих лучшее качество классификации
3. расстояние между строками кодовой матрицы максимизируется
4. Если при декодировании пользоваться оценками классификаторов, а не их ответами, учитывается качество классификации объекта бинарными классификаторами на шаге декодирования

2.2 Зависимость количества исправляемых ошибок от числа рассматриваемых задач

Пусть мы строим кодовую матрицу для задачи классификации на K классов. Пусть в композицию будут входить N классификаторов. При этом рассматривается M классификаторов ($N < M \leq 2^{K-1} - 1$).

Предположим, что количество ошибок, которое способна исправить кодовая матрица, мало зависит от M, при фиксированном N. Проверим справедливость этого утверждения при различных N, M и K.

Для этого исследуем распределение минимального расстояния в зависимости от K:

- Зафиксируем N и K;
- Для различных M оценим плотность распределения минимального межстрочного расстояния
- Для каждого M случайно выберем M двоичных векторов $\{0,1\}^{K\times 1}$, таких что среди них нет пары векторов с различными номерами, которые равны или в сумме равны единичному или нулевому вектору.
- Для каждого M из отобранных столбцов построим 5000 матриц $\{0,1\}^{K\times N}$. В каждой из матриц нет двух столбцов с различными номерами, которые равны или в сумме равны единичному или нулевому столбцу.
- Для каждой матрицы найдем минимальное расстояние между строками

- по полученным данным оценим распределение минимального межстрочного расстояния в зависимости от M, N и K.

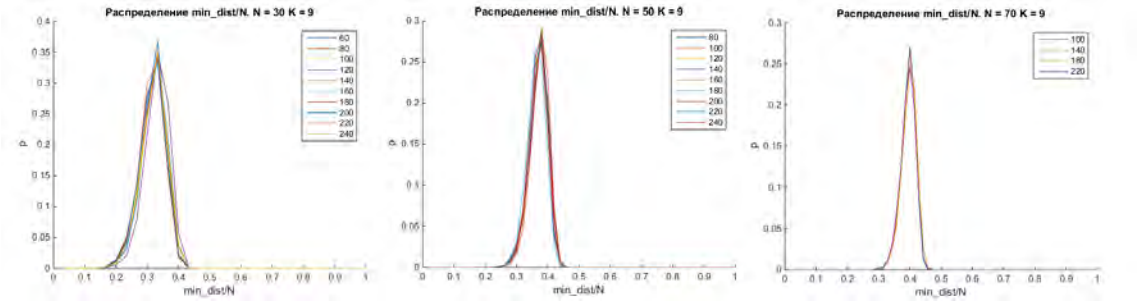

Рис. 1: Графики распределения минимального межстрочного расстояния при различных M

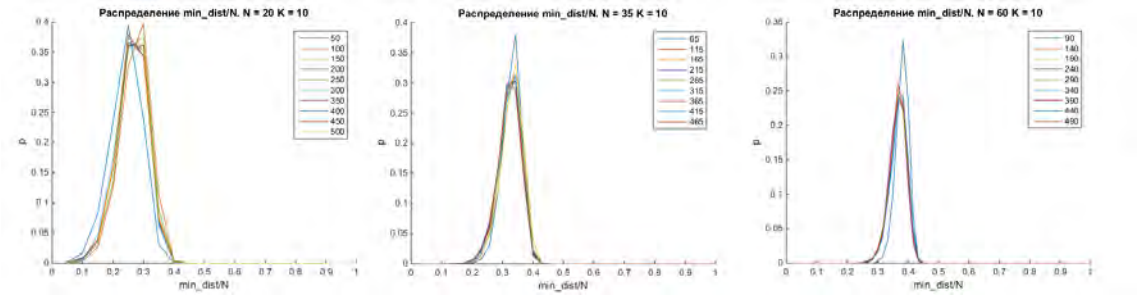

Рис. 2: Графики распределения минимального межстрочного расстояния при различных M

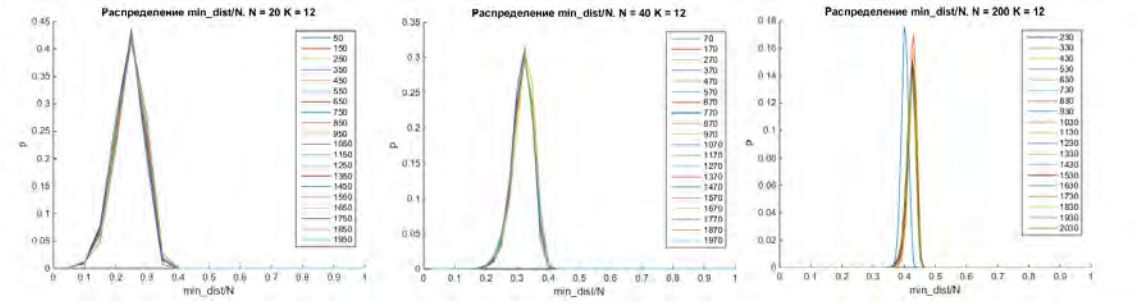

Рис. 3: Графики распределения минимального межстрочного расстояния при различных M

Распределение почти не зависит от M. Дисперсия мала. Выбросы отсутствуют. Серьезное уменьшение минимального расстояния произошло только при $N = 200, M = 230$.

Во многих случаях предположение верно. Перед применением алгоритма стоит проверить выполнение предположения о влиянии M на число исправляемых ошибок.

2.3 Качество бинарной классификации в зависимости от задачи

В работе утверждалось, что выбранное для построения композиции семейство классификаторов не может одинаково хорошо решать все бинарные задачи. Подтвердим это утверждение экспериментом на реальных данных.

Алгоритмы тестировались на базе данных MNIST [7].
Количество классов: 10
Количество объектов: 60000
Размерность данных: 784 (28x28)
Пропусков в данных нет

Объекты - изображения рукописных цифр в градациях серого. Классы сбалансированы по количеству элементов.

На 50000 объектов MNIST были обучены машины опорных векторов(SVM) с гауссовским ядром. Для каждого SVM была подсчитана доля ошибок на скользящем контроле (5-fold CV [12]).

Полученные доли ошибок классификаторов были отсортированы по возрастанию.

Качество классификации сильно зависит от задачи - лучший алгоритм со-

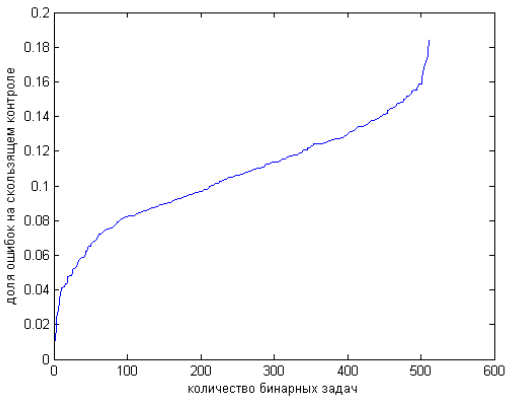

Рис. 4: Отсортированные ошибки на скользящем контроле для бинарных задач

вершает ошибок в 9 раз меньше, чем худший. Важно отбирать в композицию хорошо обученные алгоритмы.

Оценить качество классификации, не обучая соответствующий классификатор, невозможно.[12] Однако, можно увидеть, какие классификаторы обучены лучше всего.

Для этого построим следующий график: по оси x отложим номер бинарной задачи, по оси y отложим долю ошибок соответствующего классификатора.

На рисунке 5 номер бинарной задачи присваивался следующим образом: ind -

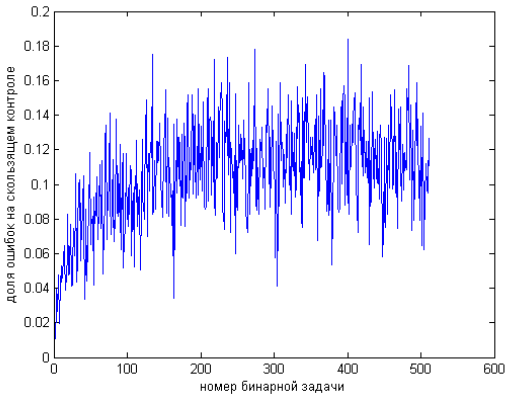

Рис. 5: Ошибка классификатора в зависимости от задачи

номер бинарной задачи. $ind \in mathbb(N), ind \leq 511$
m^ind - соответствующий столбец кодовой матрицы
не ограничивая общности предположим, что в столбцах кодовой матрицы не более пяти единиц.

$m^1 = (1,0,0,0,0,0,0,0,0,0)^T$
если число единиц в столбце m^{ind_1} меньше, чем в столбце m^{ind_2}, то $ind_1 < ind_2$
если число единиц в столбцах m^{ind_1} и m^{ind_2} одинаково и двоичное число $\overline{1m^{ind_1}}_2 > \overline{1m^{ind_2}}_2$, то $ind_1 < ind_2$

Обучив M_0 классификаторов, важно построить аналогичные графики. Они помогут определить диапазон изменения параметров M и N.

3 Эксперименты на данных

Алгоритм для сравнения

Обратим внимание на рисунок 5. На нем видно, что первые 10 алгоритмов (они соответствуют задачам метода "один против всех") наиболее просты для рассмотренного семейства алгоритмов. Доля ошибок на скользящем контроле классификаторов метода "один против всех"в разы ниже средней. Метод "один против всех"задает очень высокую планку.

В качестве метода для сравнения возьмем алгоритм "один против всех".

Если в качестве ответа композиции принять сигма - функцию от расстояния до разделяющей поверхности и измерять расстояние от ответа до строк кодовой матрицы при помощи расстояния Евклида, получим 0.1063 ошибок на тестовой

выборке.

Сравним результаты композиций ECOC, декодированных при помощи обобщенного алгоритма декодирования и алгоритма с использованием ответов классификаторов. На графике видно, что доля ошибок при использовании обобщен-

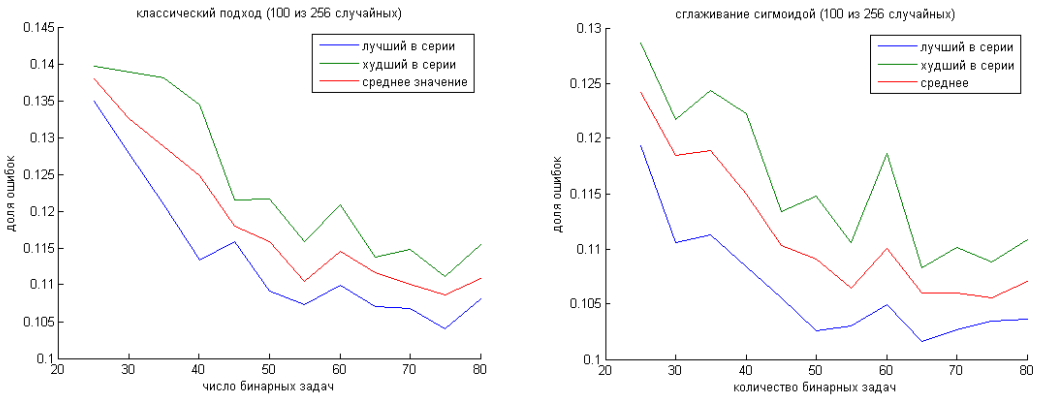

Рис. 6: $M_0 = 256, M = 100, N$ отложено по оси X

ного алгоритма декодирования меньше. Изучим графики подробнее.

1. Разница между лучшим и худшим запусками ощутима. Это объясняется тем, что в матрицы входят разные столбцы, соответствующие задачам с разной сложностью.

2. С ростом N качество классификации увеличивается.

3. Удалось достичь качество метода "один против всех".

Запустим алгоритм подбора параметров при $M_0 = 130$ Получим $M = 90, N = 90$, доля ошибок равна 0.0994.
Сетка:
$M = \{50, 70, 90, 110, 130\}$
$N = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130\}$
Превзойти метод "один против всех"удалось.

Запустим алгоритм настройки параметров при $M_0 \in \{50, 70, 90, 190\}$ Получим следующие доли ошибок на тестовой выборке:

M_0	50	70	90	190
доля ошибок	0.1086	0.1050	0.1030	0.1006

Изменение M_0 слабо влияет на качество распознавания.

Настройка параметров кодовой матрицы позволила улучшить результаты метода "один против всех". Не смотря на то, что задача отделения одного класса от остальных была самой легкой подбор параметров кодовой матрицы позволил уменьшить число ошибок на тестовой выборке. Эти результаты показывают важность настройки параметров композиции по данным и демонстрируют работоспособность предложенного алгоритма настройки параметров. Если в некоторой задаче не удается качественно отделить один класс от остальных, метод настройки параметров кодовой матрицы позволит достигнуть приемлемого качества классификации.

4 Заключение

4.1 Выводы

Метод ECOC - стандартный метод решения многоклассовых задач. Многие методы классификации можно свести к методу ECOC.

Важно уметь настраивать по данным длину кодовой матрицы, а также при построении матрицы отдавать предпочтение столбцам, которым соответствуют более простые бинарные задачи.

На этапе декодирования стоит учитывать не только ответы бинарных классификаторов, но и отступы классифицируемых объектов.

ECOC - композиция алгоритмов. Ее параметры можно оценить по данным. Точная настройка параметров - задача, сложность которой экспоненциально зависит от количества классов, однако, существуют эффективные приближенные решения.

4.2 Что выносится на защиту

1. Алгоритм настройки параметров композиции ECOC
2. Обоснование алгоритма
3. Метод обобщения многоклассового SVM на произвольную кодовую матрицу.

5 Список литературы

1. T. Windeatt and G. Ardeshir, "Boosted ECOC Ensembles for Face Recognition," Proc. Int'l Conf. Visual Information Eng., pp. 165-168, 2003.

2. R. Ghani, "Combining Labeled and Unlabeled Data for Text Classification with a Large Number of Categories," Proc. Int'l Conf. Data Mining, pp. 597-598, 2001

3. J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas, "Face Verification Using Error Correcting Output Codes," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 755-760, 2001.

4. O. Pujol, S. Escalera, and P. Radeva, "An Incremental Node Embedding Technique for Error Correcting Output Codes," Pattern Recognition, to appear.

5. O. Pujol, P. Radeva, and J. Vitria', "Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes," IEEE Trans. Pattern Analysis

6. T. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes," J. Artificial Intelligence Research, vol. 2, pp. 263-286, 1995.

7. http://yann.lecun.com/exdb/mnist/

8. Shannon C.E. A Mathematical Theory of Communication // Bell System Technical Journal. — 1948. — Т. 27. — С. 379-423, 623–656.

9. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. — М.: Мир, 1976.

10. Gallager, R. G., Low Density Parity Check Codes, Monograph, M.I.T. Press, 1

11. D. H. Wolpert. Stacked generalization. Neural Networks, 5:214–259, 1992.

12. К. В. Воронцов. Математические методы обучения по прецедентам (теория обучения машин). Москва, 2011.