Московский Государственный Университет
им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Дипломная работа:

«Применение комбинаторных оценок обобщающей способности для повышения качества метрических алгоритмов классификации»

Выполнил студент 517 группы:
Колосков А. О.

Научные руководители:
чл.-корр. РАН, д.ф.-м.н. Рудаков К.В.
к.ф.-м.н. Воронцов К.В.

Москва
2005

Содержание

Аннотация

Работа выполнена в рамках комбинаторной теории качества восстановления зависимостей по прецедентным данным, предложенной К. Воронцовым. Рассматриваются алгоритмы классификации, основанные на методе ближайших соседей. Предлагается новый метод отбора опорных объектов, основанный на понятии профиля компактности и комбинаторных формулах для эффективного вычисления функционала скользящего контроля. Показано, что данный метод разделяет обучающие объекты на три категории: шумовые выбросы, неинформативные (периферийные) объекты и опорные объекты. Исключение шумовых и периферийных объектов из обучающей выборки повышает обобщающую способность метода ближайших соседей и существенно снижает затраты времени на классификацию новых объектов.

1 Введение

Метрические алгоритмы классификации используются для решения широкого класса прикладных задач, в которых естественным образом возникает понятие сходства объектов. В основе этих алгоритмов лежит гипотеза компактности – предположение о том, что схожие объекты, как правило, лежат в одном классе. Самым непосредственным выражением гипотезы компактности является алгоритм k ближайших соседей (kNN), относящий распознаваемый объект к тому классу, которому принадлежит большинство из k ближайших к нему объектов обучающей выборки.

Известно большое количество модификаций алгоритма k ближайших соседей, направленных на повышение качества классификации и обогащение алгоритмической модели путем введения дополнительных параметров. Одна из возможных модификаций заключается в том, чтобы отобрать небольшое подмножество опорных объектов (эталонов), и сравнивать распознаваемые объекты только с этими эталонами. Отбор опорных объектов имеет ряд важных преимуществ:

- Повышается качество классификации за счет отбрасываемых объектов, заданных со значительными искажениями информации (шумовых выбросов).
- Снижаются затраты машинного времени на поиск ближайших соседей при классификации новых объектов. Это особенно важно для прикладных задач со сверхбольшими объемами данных.
- Формируется сжатое описание обучающей выборки. Оставшиеся опорные объекты можно предъявлять в качестве типичных представителей классов на стадии интерпретации решений, выдаваемых алгоритмом.

Заметим, что отбрасывание некоторых объектов вовсе не означает потерю информации, поскольку в процессе отбора задействуется вся обучающая выборка.

Известно немало методов классификации, в которых алгоритм строится не по всей обучающей выборке, а только по небольшому ее подмножеству. Классический пример такого метода — машины опорных векторов Вапника (Support Vectors Machines, SVM). Опорными векторами в SVM оказываются объекты, непосредственно примыкающие к границе разделения классов. Этот принцип отбора опорных объектов неоднократно подвергался критике, поскольку граничными объектами в условиях неполных и неточных данных часто становятся шумовые выбросы. Идея брать в качестве опорных объекты, «слегка отодвинутые» от границы классов, лежит в основе метода релевантных векторов Типпинга (Relevance Vectors Machines, RVM).

Среди метрических алгоритмов отбор опорных объектов осуществляют СТОЛП и λ-СТОЛП [5]. Они основаны на жадной стратегии последовательного добавления опорных объектов. На каждом шаге алгоритма опорным объявляется тот объект, для которого локальная плотность объектов чужих классов в его ближайшем окружении максимальна. В результате опорными становятся пограничные объекты, аналогично тому, как это происходит в SVM.

Для реализации метрического алгоритма с более разумным механизмом отбора опорных объектов, подобным RVM, необходим гораздо более тонкий критерий, способный обоснованно решать, как далеко от границы классов должны отстоять опорные объекты.

В данной работе предлагается критерий, основанный на эффективных комбинаторных формулах скользящего контроля и понятии профиля компактности выборки [4]. Разработанный алгоритм выделения опорных объектов работает в противоположном направлении по сравнению с алгоритмами класса СТОЛП. Начиная с полной выборки, он сначала исключает шумовые объекты, отбрасывание которых приводит к улучшению функционала. Затем исключаются неинформативные периферийные объекты, отбрасывание которых не изменяет значение функционала или приводит к его несущественному ухудшению. Процесс останавливается, когда остаются объекты, исключение которых заметно ухудшает функционал. В качестве побочного результата возникает деление обучающих объектов на три типа: шумовые, периферийные и опорные.

Комбинаторные функционалы скользящего контроля характеризуют качество обучения (обобщающую способность) алгоритма [3]. Поэтому есть основания полагать, что предложенный алгоритм отбора опорных объектов приводит к более качественной классификации. Для проверки этой гипотезы в работе проведены эксперименты на модельных данных. Предложенный алгоритм показал заметно лучшие результаты при классификации тестовых данных по сравнению с алгоритмами СТОЛП и λ-СТОЛП.

Таким образом, основным результатом применения комбинаторной формулы для оценки функционала полного скользящего контроля является то, что она одинаково хорошо подходит как для исключения шумовых объектов, так и для сокращения множества прецедентов, являясь при этом эффективно вычислимым, точным значением функционала.

Актуальность данного исследования вызвана появлением новых прикладных задач классификации со сверхбольшим числом объектов и представлением исходных данных в виде попарных оценок сходства объектов.

Новизна Идея применения комбинаторных формул скользящего контроля и профиля компактности выборки для выбора опорных объектов является новой.

1.1 Задача обучения по прецедентам

Имеется множество *объектов X*, множество *ответов Y* и множество *A* отображений из *X* в *Y*, элементы которого будем называть алгоритмами, имея в виду, что они являются эффективно вычислимыми функциями. Предполагается, что существует отображение $y^* : X \to Y$, не обязательно принадлежащее A, называемое *восстанавливаемой зависимостью*. Значение отображения $y^*(x)$ известны только на конечном множестве объектов $X^l = \{x_1,...,x_l\}$, называемом *обучающей выборкой.* Обозначим ответ на *i*-ом объекте обучающей выборки через $y_i = y^*(x_i)$.

Задача обучения по прецедентам заключается в том, чтобы, используя только обучающую выборку, построить алгоритм $a^* \in A$, удовлетворяющий трем требованиям.

Во-первых, он должен выдавать на объектах обучающей выборки заданные ответы, т.е. $a^*(x_i) = y_i, i = 1, ..., l$. Равенство здесь может пониматься как точное или приближенное в зависимости от конкретной задачи. Требования такого вида называют *локальными ограничениями*, подчеркивая, что они связаны с конечным числом обучающих объектов и допускают эффективную проверку за конечное число шагов.

Во-вторых, на алгоритм a^* могут накладываться дополнительные ограничения общего характера, которым он должен удовлетворять как отображение, действующее их *X* в *Y*. Например, это могут быть ограничения симметричности, непрерывности, гладкости, монотонности, и т.д., а также их сочетания. Требования такого вида называют *универсальными ограничениями*, подчеркивая. Что они не зависят от конкретной обучающей выборки и относятся к отображению «в целом». Как правило, они не допускают эффективной конечной проверки и учитываются в самой конструкции алгоритма на этапе его разработки. В общем случае универсальные ограничения выражаются условием $a^* \in A_u$, где A_u – заданное подмножество алгоритмов, определяемое спецификой задачи.

В-третьих, искомый алгоритм a^* должен обладать *способностью к обобщению* (generalization performance), то есть приближать восстанавливаемую зависимость y^* не только на объектах обучающей выборки, но и на всем множестве *X*. Данное требование можно формализовать с помощью различных функционалов качества.

Процесс построения по обучающей выборки искомого алгоритма принято называть *обучением*, а его применение для классификации новых объектов – *распознаванием*. Объекты из обучающей выборки называют *прецедентами*.

Определение 1. *Частота ошибок* алгоритма a на выборке $X^p = \{x_1, ..., x_p\} \subset X$ есть

$$\nu(a, X^p) = \frac{1}{p}\sum_{i=1}^{p} I(x_i, a(x_i)),$$

где *I(x,y)* – индикатор ошибки, принимающий значение *1*, если ответ у является ошибочным для объекта *x*, и *0* в противном случае. Выбор индикатора существенно зависит от конкретной задачи, в первую очередь от природы множества *Y*. В задачах классификации Y – конечное множество, обозначающее номера классов объектов.

Определение 2. *Методом обучения* называется отображение μ, которое произвольной конечной выборки X^l длины l ставит в соответствие определенный алгоритм $a = \mu(X^l)$. То есть метод μ строит алгоритм a по обучающей выборки X^l. Будем полагать, что метод μ строит алгоритмы, выбирая их из некоторого семейства алгоритмов $A \subseteq A_u$.

Алгоритм a называется *корректным* на выборке X^l, если $\nu(a, X^l) = 0$.

Метод μ называется *корректным* на выборке X^l, если алгоритм $\mu(X^l)$ корректен на X^l, то есть $\nu(\mu(X^l), X^l) = 0$.

1.2 Функционалы скользящего контроля

Малая частота ошибок на обучающей выборке $\nu(\mu(X^l), X^l)$ в общем случае не гарантирует, что построенный алгоритм будет редко ошибаться на остальных объектах множества *X*. Обобщающую способность метода возможно оценить только по данным, не входящим в состав обучения. Для этого вводятся функционалы качества, основанные на принципе скользящего контроля.

Пусть l и q – произвольные фиксированные числа, $L = l + k$, и задана выборка $X^L = \{x_1, ..., x_L\}$. Обозначим через (X_n^l, X_n^k), *n=1, ..., N* всевозможные разбиения выборки X^L на обучающую и контрольную подвыборки длиной l и k соответственно.

В зависимости от способа формирования множества разбиений различают несколько разновидностей скользящего контроля.

Если множество разбиений одноэлементное, говорят об оценке по отдельной тестовой выборке (hold-out estimate)

$$Q_{HO}(\mu, X^l, X^k) = \nu(\mu(X^l), X^k).$$

Недостаток функционала Q_{HO} в том, что он существенно зависит от способа разбиения выборки на обучение и контроль.

Довольно распространенным является также *функционал k-fold скользящего контроля.* Для вычисления этого функционала исходная выборка X^L разделяется на *k* подмножеств равной мощности. Каждое из таких подмножеств используется в качестве контрольной выборки алгоритма, обученного на оставшихся *k-1* подмножествах, для вычисления частоты ошибки этого алгоритма. Результирующей значение функционала получается как среднее значение от частоты ошибок каждого из *k* таких алгоритмов.

Однако наиболее точный результат дает *функционал полного скользящего контроля* (complete cross-validation), в котором учитываются все возможные разбиения исходной выборки X^L на обучающую и контрольную подвыборки длины *l* и *k* соответственно. Число таких разбиений равно $N = C_L^l$. Поэтому точное выражение функционала полного скользящего контроля имеет следующий вид:

$$Q_c^{lk}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N}\nu(\mu(X_n^l), X_n^k).$$

Недостаток функционала Q_c в том, что его практически невозможно вычислить даже для небольших *k*. Известны теоретические верхние оценки данного функционала [4], аналогичные оценкам равномерной сходимости в теории Вапника-Червоненкиса [2]. Однако они сильно завышены, что препятствует их непосредственному практическому применению. В отдельных частных случаях путем аналитических преобразований удается получить точное выражение данного функционала. Эффективно вычислимые формулы известны для алгоритма *k* ближайших соседей и некоторых его модификаций.

Если используются все разбиения с контрольной выборкой единичной длины, то говорят об оценке по одному отделяемому объекту (leave-one-out estimate, LOO), которая является частным случаем предыдущего функционала при *k = 1*:

$$Q_{LOO}(\mu, X^L) = \frac{1}{L}\sum_{i-1}^{L}\nu(\mu(X^L \setminus \{x_i\}), \{x_i\}) \ .$$

1.3 Метод *k* ближайших соседей и профиль компактности

При решении задач распознавания образов обучающая выборка каждого из *k* классов представлена конечным число *m* объектов x_i *(i = 1, ..., m)* этого класса. Сведений о законах распределения и их параметрах объектов рассматриваемых классов нет, не известна степень представительности обучающей выборки. Также нет представления об априорной

вероятности появления разных классов и о стоимости ошибок распознавания. Поэтому совершено очевидно, что для применения того или иного алгоритма нужно к имеющейся объективной информации добавить ряд субъективно выбираемых предположений или гипотез. Этот этап применения эвристических гипотез имеет место во всех случаях решения реальных задач распознавания образов.

Одна из давно используемых эмпирических гипотез, известная в литературе по распознаванию образов под именем *гипотезы компактности*, состоит в том, что объекты одного класса обычно отражаются в признаковом пространстве в геометрически близкие точки, образуя «компактные» сгустки. Несмотря на простоту этой гипотезы, она лежит в основе большинства алгоритмов не только распознавания образов, но и других задач анализа данных. Мера компактности может быть любой: она может характеризоваться средним расстоянием от центра тяжести класса до всех его объектов, средней длиной ребра полного или ребра кратчайшего незамкнутого пути, соединяющего точки одного класса или максимальным расстоянием между двумя точками класса и т.д.

Существует большое количество эвристических алгоритмов распознавания образов, одним из самых распространенных из которых является алгоритм *k ближайших соседей* (*k*NN). Этот алгоритм основывается всего лишь на одной эмпирической гипотезе – гипотезе локальной компактности, из которой следует, что в малой ε-окрестности от объекта *i*-го класса обучающей выборки могут появляться объекта только того же *i*-го класса. Причем чем ближе объект *q* из контрольной выборки находится к обучающему объекту *i*-го класса, тем с большей вероятностью отнесение точки *q* к *i*-му классу будет правильным.

Для формализации понятия «сходства» вводится функция расстояния или метрика $\rho(x,x')$ в пространстве объектов *X*. Поэтому алгоритмы, основанные на анализе сходства объектов, называют также метрическими алгоритмами. К метрическим алгоритмам классификации относятся: метод *k* ближайших соседей, метод потенциальных функций [1], метод парзеновского окна, сети радиальных функций (Radial Basis Functions, RBF), и многие другие.

Рассмотрим метод обучения μ, который строит алгоритм первого ближайшего соседа $a = \mu(X^l)$, работающий следующим образом:

$$a(x) = y^*(\arg\min_{x' \in X^l} \rho(x,x')) \text{ для всех } x \in X.$$

То есть в качестве ответа алгоритма для распознаваемого объекта принимается номер класса ближайшего к нему объекта из обучающей выборки. Точное выражение функционала полного скользящего контроля Q_{CCV} для алгоритма *k* ближайших соседей и некото-

рых его модификаций было найдено в [6]. Авторы этой работы ставили перед собой целью вывод эффективных вычислительных формул. В работе [4] этот результат был использован для введения специальной векторной характеристики выборки — *профиля компактности*, выражающей априорную информацию о компактности классов.

Для каждого объекта x_i, $i=1,...,L$ выборки X^L расположим остальные $L-1$ объектов в порядке возрастания расстояния до x_i, пронумеровав их двойными индексами:

$x_i = x_{i0}; x_{i1}, x_{i2}, ..., x_{iL-1}$.

Объекты этой последовательности удовлетворяют следующему условию:

$$0 = \rho(x_i, x_{i0}) \le \rho(x_i, x_{i1}) \le ... \le \rho(x_i, x_{iL-1}).$$

Обозначим через $r_m(x_i)$ ошибку, возникающую при замене известной классификации объекта x_i на ответ алгоритма ближайшего соседа на его m-ом соседе:

$r_m(x_i) = I(x_i, y^*(x_{im}))$; $i = 1, ..., L$; $m = 1, ..., L-1$.

Определение 3. Профилем компактности выборки X^L называется функция $P(m)$, выражающая долю объектов выборки, для которых правильный ответ не совпадает с правильным ответом на m-ом соседе:

$$P(m) = \frac{1}{L}\sum_{i=1}^{L} r(x_i); \quad m = 1,...,L-1.$$

Теорема 1. Для задачи классификации методом ближайшего соседа справедливо следующее точное выражение функционала полного скользящего контроля Q_c^{lk}:

$$Q_c^{lk}(\mu, X^L) = \sum_{m=1}^{k} P(m)\frac{C_{L-1-m}^{l-1}}{C_{L-1}^{l}}. \qquad (1)$$

Отметим следующие свойства профиля компактности.

1. При фиксированной длине контрольной выборки k функционал зависит только от начального отрезка профиля $P(1), ..., P(k)$.

2. Комбинаторный множитель $\widetilde{C}(m) = \frac{C_{L-1-m}^{l-1}}{C_{L-1}^{l}}$ убывает с ростом m быстрее геометрической прогрессии.

3. Чтобы обеспечить малое значение функционала полного скользящего контроля (CCV), а, следовательно, и малое значение частоты ошибок алгоритма ближайшего соседа при всевозможных разбиениях исходной выборки на обучающую и контрольную, достаточно потребовать, чтобы профиль компактности принимал малые значения при малых m. Но это и означает, что близкие объекты лежат преимущественно в одном классе. При больших m рост профиля компактности компенсируется комбинаторным множителем, по-

этому далекие объекты могут располагаться как угодно. Основываясь на определении профиля компактности для исходной обучающей выборки можно сделать вывод о пригодности применения алгоритма ближайшего соседа для распознавания новых объектов. Таким образом, малое значение профиля компактности при малых m свидетельствует о компактном расположении классов исходной обучающей выборки. Это дает основание полагать, что применение алгоритма ближайшего соседа обеспечит высокое качество распознавания. Рис. 1 показывает, что чем проще задача (чем более адекватна введенная мера сходства объектов), тем ниже лежит начальный отрезок профиля компактности.

4. Эксперименты на модельных данных свидетельствуют также о том, что функционал CCV не сильно зависит от доли объектов $\frac{k}{L}$, оставляемых для контроля (см. нижний ряд графиков на рис. 1). Резкое ухудшение происходит только, когда в обучении остается критически мало объектов. Эти наблюдения позволяют выдвинуть предположение, что всего несколько контрольных объектов уже достаточно для адекватного оценивания качества обучения (гипотеза адекватности функционала LOO). Заметим, что при малых k вычисление функционала существенно более эффективно.

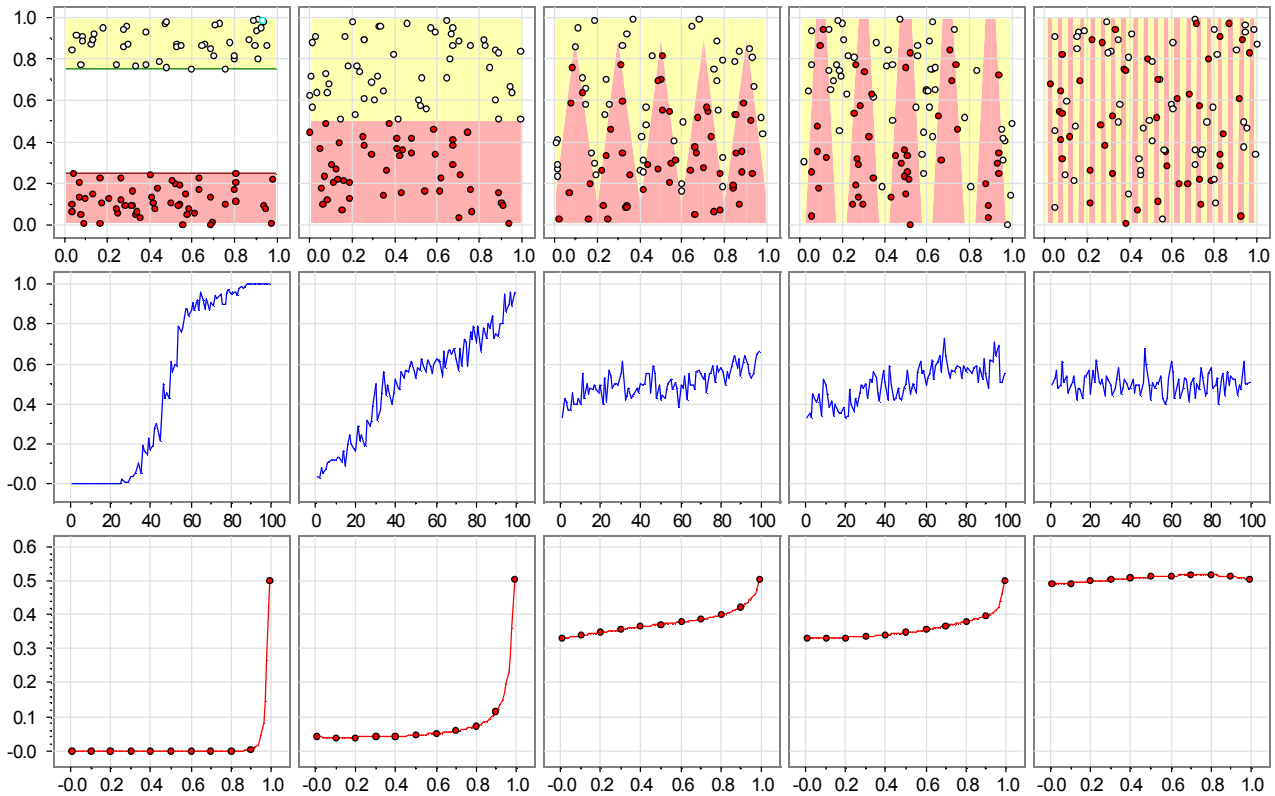

Рис. 1: Верхний ряд: пять модельных задач, в порядке возрастания трудности. Средний ряд: профили компактности. Нижний ряд: зависимости Q_c от отношения k/L.

1.4 О способах улучшения метрических алгоритмов

Основные известные модификации метода *k*NN направлены на обогащение модели алгоритмов путем введения дополнительных параметров.

1. Введение весов признаков. Введение весов признаков. Как правило, в прикладной задаче отсутствует какой-либо приоритетный способ введения метрики. Если объекты описываются набором признаков, то метрику можно построить как линейную комбинацию элементарных метрик, соответствующих отдельным признакам. Тогда коэффициенты линейной комбинации (веса метрик) становятся дополнительными параметрами, которые подлежат настройке по обучающей выборке.

2. Введение весов объектов. Обучающие объекты могут иметь различную полезность с точки зрения классификации новых объектов. Сохранение шумовых объектов (выбросов) в качестве обучающих может оказаться даже вредным и только ухудшить качество алгоритма. Вклад этих объектов в классификацию должен быть уменьшен. В общем случае более тонкое распределение весов способно повысить качество классификации.

3. Фильтрация объектов. Обучающая выборка может оказаться избыточной не только по причине наличия выбросов. Во многих современных прикладных задачах данные имеют сверхбольшой объём — десятки тысяч объектов и более. В таких случаях выборка содержит слишком много похожих объектов, практически бесполезных для классификации, но значительно увеличивающих время поиска ближайших соседей и объём хранимой информации. Поэтому имеет смысл ставить задачу выделения оптимального подмножества опорных объектов, действительно необходимых для классификации. Примером этого подходя являются алгоритмы СТОЛП и λ-СТОЛП [5].

1.5 Постановка задачи

В данной работе рассматривается задача фильтрации объектов для метрических алгоритмов классификации, в частности, для алгоритма ближайших соседей. Для отбора опорных объектов используются комбинаторные оценки обобщающей способности, основанные на анализе профиля компактности.

Основные цели работы

- Проверить гипотезу, что непосредственное использование комбинаторных оценок обобщающей способности для выбора подмножества опорных объектов позволяет улучшить качество метрических алгоритмов классификации.

- Разработать и реализовать алгоритм отбора опорных объектов, основанный на оптимизации профиля компактности.
- Провести эксперименты на модельных данных по сравнению данного алгоритма с эвристическими методами отбора опорных объектов, в частности, с алгоритмами СТОЛП и λ-СТОЛП.

2 Алгоритмы выбора опорных объектов

Существуют различные пути улучшения профиля компактности исходной выборки, основными из которых являются: синтез метрик и выделение множества опорных объектов. В этой работе подробно рассматривается второй подход, который заключается в выделение из исходной обучающей выборки подмножество объектов, обучаясь на котором, получается алгоритм с улучшенным качеством распознавания.

Основным недостатком алгоритма *k* ближайших соседей заключается в необходимости хранить все объекты из обучающей выборки и сравнивать неизвестный объект со всеми этими прецедентами. Объекты, находящиеся вблизи границы байесовского решающего правила, сильно влияют на результат применения метода *k*NN, а объекты, далекие от границы, не влияют на решение. Поэтому исключение этих невлияющих объектов позволяет уменьшить как время на распознавание неизвестного объекта, так и экономит память.

Таким образом, множество опорных объектов обеспечивает безошибочное распознавание всех объектов исходной выборки. Сложность отбора опорных объектов состоит в том, что состав прецедентов *i*-го класса зависит от того, какие других классов выбраны в качестве опорных. Из этого следует комбинаторный характер задачи, оптимальное решение которой в общем случае требует полного перебора всех вариантов. Если число объектов исходной выборки L, число объектов в первом классе равно m_1, а во втором m_2, то число возможных вариантов выбора по одному прецеденту для каждого класса равно $m_1 \cdot m_2$. Если же оставлять по t прецедентов для каждого класса, то получаем $C_{m_1}^t \cdot C_{m_2}^t$ вариантов. Поэтому большинство существующих алгоритмов выбора множества опорных объектов не использует полный перебор вариантов, а использует эмпирические гипотезы для построения таких алгоритмов.

2.1 Алгоритм СТОЛП

Одним из таких эмпирическим алгоритмом отбора множества опорных объектов, который сокращает полный перебор всех вариантов, является алгоритм СТОЛП.

Рассмотрим его работу на примере с двумя классами. Сначала находятся самые « «напряженные» пограничные точки из исходной обучающей выборки. С этой целью для каждой точки определяются расстояния до ближайшей точки своего класса (r_{in}) и ближайшей точки чужого класса (r_{out}). Отношение $W = \frac{r_{in}}{r_{out}}$ характеризует величину риска для данной точки быть распознанной в качестве точки чужого класса. Среди точек каждого класса выбираются по одной точке с максимальным значением величины *W*. Эти точки заносятся в список опорных объектов(прецедентов).

Затем делается пробное распознавание всех точек обучающей выборки с опорой на прецеденты и с использованием правила ближайшего соседа: точка относится к тому классу, расстояние до прецедента которого минимально. Среди точек, распознанных неправильно, выбирается точка с максимальным значением W и ею пополняется список прецедентов, после чего повторяется процедура пробного распознавания всех точек. Так продолжается до тех пор, пока все точки обучающей выборки не станут распознаваться без ошибок.

Недостаток алгоритма в том, что он основан на вычислении эвристической оценки «напряжённости» объекта *W(x)*, не имеющей строгого обоснования. Есть основания полагать, что критерии отбора опорных объектов, построенные на основе комбинаторного функционала, выражающего обобщающую способность метода обучения, дадут лучшие результаты.

2.2 Алгоритм λ-СТОЛП

Этап отбора опорных объектов в алгоритме λ-СТОЛП состоит из следующих процедур. Для каждого из *N* классов в отдельности находится максимальное расстояние между двумя объектами, принадлежащих этому классу, и обозначается через D_A. Для каждого объекта x_i через β_i обозначается расстояние до ближайшего к нему объекту. Затем вычисляется самое большое значение характеристики локального скачка плотности τ_A.

Далее вводится понятие *функции принадлежности* объекта *q* к образу *A*. Среди всех объектов класса А находится «ближайший сосед» — объект *i*, удаленный от *q* на минимальное расстояние $d(q,i)$. Нормированное расстояние *d* между объектами *q* и *i* равно $d(q,i)/D_A$. Теперь можно определить величину $\tau^* = \frac{d(q,i)}{\beta_i}$ и нормированное значение этой величины $\tau = \tau^*/\tau_A$. Затем вычисляется характеристика $\lambda(A,q) = \tau^2 d$.

Функция принадлежности $f(A,q)$ объекта *q* к образу *A* равна: $f(a,q) = 1 - \lambda(a,q)$.

Для всех объектов *i* обучающей выборки вычисляются значения функции принадлежности к своему классу $f(A,i)$ и ко всем другим классам $f(A^-,i)$. Среди объектов каждого класса находятся точки «максимального риска», т.е. такие объекты, у которых величина $R = f(A^-,i) - f(A,i)$ имеет наибольшее значение. Эти *N* объектов заносятся в список прецедентов.

Затем применяется стратегия пошагового уменьшения максимального риска. Для этого оценивается функция принадлежности всех объектов (кроме прецедентов) к своим и чужим классам с опорой только на имеющиеся точки-прецеденты. Находится один объект, имеющий самое большое значение функции риска *R*. Этот *(N+1)*-ый объект пополняет список объектов-прецедентов. После этого процедура оценки величины *R* повторяется для всех оставшихся *(m – N – 1)* объектов и самый «рискованный» из них включается в список прецедентов. Процесс продолжается до тех пор, пока самый большой риск ($R_{\max}$) для каждого объекта быть распознанным в качестве объекта чужого класса не станет меньше заданной пороговой величины R^* (например, $R^* = 0$).

2.3 Алгоритм, основанный на оптимизации профиля компактности (*Алгоритм CCV*)

В этой работе вместо создания еще одного эвристического алгоритма была предпринята попытка использовать для отбора опорных объектов критерий минимизации функционала полного скользящего контроля, представленного в виде комбинаторной формулы. Поэтому главной задачей, поставленной в этой работе, является построить алгоритм отбора опорных объектов для метода ближайших соседей, основанный на оптимизации профиля компактности.

Сначала алгоритм производит исключение из обучающей выборки шумовых объектов, которые располагаются вблизи границы, и делают ее сильно изрезанной. Сохранение всех объектов обучающей выборки позволяет более точно классифицировать исходные объекты, но в то же время сильно изрезанная граница не отражает реальную картину расположения классов, в результате чего обобщающая способность алгоритма снижается. Второе преобразование, которое выполняет этот алгоритм над обучающей выборкой, – это исключение далеких от границы, *периферийных* объектов, не являющихся представительными и не оказывающих большого влияния на результат классификации, но вместе с тем сильно увеличивающих время распознавания объектов. Таким образом, в результате этих преобразований остаются только представительные объекты из каждого класса. Эти ос-

тавшиеся объекты «среднего слоя» называются в данной работе *опорными*. Опишем подробно процесс отбора опорных объектов.

Селекция объектов производится на основе комбинаторной формулы для функционала качества, основанного на полном скользящем контроле. Эффективная для вычислений формула, полученная в [6], позволяет быстро вычислить этот функционал для исходной выборки и для выборки, полученной из исходной путем удаления некоторых объектов. На основе определение профиля компактности, введенного в [4], функционал качества запишется в виде: $Q_c^{lk} = \sum_{m=1}^{k} P(m)\widetilde{C}(m)$, где $P(m)$ – профиль компактности для *m-го* соседа, $\widetilde{C}(m)$ – комбинаторный множитель.

Для каждого объекта x_e обучающей выборки вычислим разность между значением функционала полного скользящего контроля двух выборок: S – исходной обучающей выборки и S'_e – исходной выборки, из которой удален данный объект x_e. Обозначим это приращение через $\Delta Q_c(x_e) = \sum_{m=1}^{k} \left(P(m) - P'(m)\right)\widetilde{C}(m)$, где $P(m)$ и $P'(m)$ – профили компактности соответственно выборок S и S'_e.

Процедуру исключения какого-либо объекта из обучающей выборки, которую обозначим для дальнейшего использования (*), заключается в следующем: выбранный объект исключается только из обучения, т.е. он не используется для классификации других прецедентов, однако попадает в контрольную выборку и результат его классификации используется для формирования значения функционала Q_c^{lk}.

Для уменьшения объема вычислений представим данной приращение функционала в виде вектора разности соответствующих компонент профилей компактности двух выборок. Таким образом, будет построен набор векторов $\delta(e) = \left(\delta_1(e), ..., \delta_k(e)\right)$, *i*-ая компонента которого равна $\delta_i(e) = P(i) - P'(i)$ разности профилей компактности выборки S и S'_e. Затем эти вектора сортируются по убыванию. Здесь существуют различные варианты выбора оператора сравнения двух векторов приращения профиля компактности $\delta(e)$. Например, из двух векторов большим можно считать тот, у которого начальные компоненты превосходят, аналогичные компоненты и второго вектора.

На основе введенного выше определения вектора приращения компонентов профилей компактности $\delta(e)$, приращение функционала *CCV* запишется в следующем виде:

$$\Delta Q_c(x_e) = \sum_{m=1}^{k} \delta_m(e)\widetilde{C}(m),$$

где k – число объектов в контрольной выборке для алгоритма *CCV*.

На первом этапе работы алгоритма, после сортировки векторов $\delta(e)$, первые объекты, у которых разность приращения функционала положительна, т.е. вектор *δ(e)* больше вектора с нулевыми компонентами, исключаются из обучающей выборки. Исходя из этих рассуждений, введем понятие *шумового* объекта.

Определение 4. *Шумовым*, называется такой объект исходной обучающей выборки, после удаления которого значение комбинаторного функционала полного скользящего контроля уменьшается. На основе введенных выше обозначений x_e – шумовой объект, если $\delta(e) > 0$.

В результате работы алгоритма на первом этапе настройки множества прецедентов, происходит исключение всех шумовых объектов, что приводит, как показали эксперименты, к выравниванию границы, разделяющей классы, повышая тем самым обобщающую способность алгоритма, в смысле функционала полного скользящего контроля.

После удаления первой серии шумовых объектов, для которых $\delta(e) > 0$, можно попробовать исключить шумовые объекты из вновь сформированного множества прецедентов. Аналогично предыдущему шагу происходит пересчет и сортировка векторов $\delta(e)$, вычисленных только для оставленных на предыдущем шаге объектов. Те из прецедентов, после удаления которых значения функционала уменьшилось по сравнению с вычисленным на основе, отобранных на предыдущем шаге, объектов, подлежат аналогичному исключению из множества прецедентов. Этот процесс повторяется до тех пор, пока на очередном шаге вычисления приращения функционала оказалось, что дальнейшее удаление любого объекта из оставшихся прецедентов, приводит только к увеличению Q_c. Тем самым можно считать, что все шумовые объекты исключены.

На втором этапе обора объектов, можно продолжить дальнейшей исключение объектов с минимальным отрицательным вектором приращения. Для определения таких объектов, необходимо задать некоторое малое значение ε, на которое допускается увеличение значения функционала Q_c^{lk}. Как показано на следующем рис. объекты, после удаления которых функционал качества незначительно увеличивается, располагаются далеко от границы, являясь *периферийными*, и не влияют на разделение классов.

Определение 5. Объект называется *периферийным*, если после его исключения из обучающей выборки в смысле (*), значение функционала Q_c^{lk} увеличивается не более, чем на малое, заранее выбранное значение ε. Во введенных ранее обозначениях это означает, что приращение функционала удовлетворяет следующим неравенствам: $-\varepsilon \le \Delta Q_c \le 0$.

Для повышения устойчивости процедуры отбора опорных объектов, можно проводить пересчет векторов $\delta(e)$ после исключения очередной группы периферийных объектов. Исключения таких объектов не приводит к ухудшению качества распознавания, так как они не оказывают существенного влияния на результат. Отсутствие этих объектов легко компенсируется заменой их ответа на ответ оставленных опорных объектов. Процедура исключение периферийных объектов, аналогично первому этапу, повторяется до тех пор, пока на очередном шаге исключение любого из оставшихся прецедентов приведет к увеличению функционала более, чем на ε, то есть $\Delta Q_c(e) < -\varepsilon, \quad \forall x_e \in$ *оставшемуся множеству прецедентов*. В результате данный этап алгоритма позволяет исключить все периферийные объекты из исходной обучающей выборки, не оказывающие влияния на качество классификации алгоритма. Таким образом, вектор $\delta(e)$ приращения профилей компактности выборок S и S'_e можно считать мерой представительности прецедента, чем меньше этот вектор, тем больший вклад в распознавание вносит данный объект x_e.

После проведения первого и второго этапа исключения шумовых и периферийных объектов, оставшееся множество объектов является *опорным*. В него включен минимальный набор объектов, необходимый для достижения максимального качества работы алгоритма.

2.4 Формальное описание алгоритма CCV на псевдокоде

Процедура П1(in S).

1. Для каждого объекта x_i, $i=1,\ldots,L$ вычислим вектор приращения компонент профиля компактности исходной выборки S и выборки $S_i' = S \setminus \{x_i\}$ с исключенным, в смысле (*), объектом x_i.
2. Упорядочиваем по убыванию вектора приращения $\delta(i)$, в соответствии с заранее выбранным оператором сравнения таких векторов.

Алгоритм отбора опорных объектов

1. Для каждого объекта x_i, $i=1,\ldots,L$ исходной обучающей выборки X^L расположим остальные $L-1$ объектов в порядке возрастания расстояния до x_i, получим упорядоченный вектор $x_i = x_{i0}; x_{i1}, x_{i2}, \ldots, x_{iL-1}$.
2. *Исключение шумовых объектов:*

   $\widetilde{S} := S$;

   repeat

   $S := \widetilde{S}$;

   выполняем **процедуру П1**(S);

   после этого формируем новую обучающую выборку $\widetilde{S} = \bigcap_{i:-\varepsilon\le\delta(i)\le 0} S_i'$, из которой исключаются шумовые объекты, уменьшающих функционал CCV;

   until $\widetilde{S} \neq \varnothing$.
3. *Исключение периферийных объекты:*

   repeat

   $S := \widetilde{S}$;

   выполняем **процедуру П1**(S);

   после этого формируем новую обучающую выборку $\widetilde{S} = \bigcap_{i:-\varepsilon\le\delta(i)\le 0} S_i'$, из которой исключаются периферийные объекты;

   until $\widetilde{S} \neq \varnothing$.
4. S – результирующее множество опорных объектов.

Сложность этого алгоритма равна $O(\log(L^3))$.

3 Исследование параметров алгоритма CCV

В описанном выше алгоритме CCV существенны следующие момента:

1) Способ сравнения векторов приращения.
2) Критерий остановки отброса, объектов незначительно увеличивающих функционал.
3) Последовательность исключения шумовых объектов.

3.1 Способ сравнения векторов приращения

При сравнении векторов приращения функционала необходимо выбрать число первых значащих компонент, на основе которых будет определяться знак приращения. При проведении экспериментов выяснилось, что существенными являются только первые несколько компонент. Для выяснения этого числа проведем следующие рассуждения.

Комбинаторные множители $\widetilde{C}(m)$ резко убывают с ростом числа m, поэтому получив значительное положительное приращение функционала уже по первым значениям профиля компактности, дальнейшее рассмотрение остальных его значения для остальных соседей является необязательным и только приводит к возрастанию числа вычислений, при этом не оказывая влияния на результат. И наоборот, если удаление какого-либо объекта привело к увеличению профиля компактности для первых $m=1,2,3\ldots$, то даже в самой лучшей последующей ситуации в плане уменьшения функционала, т.е. $\delta_i(e) = 1$, при больших i, в результате все равно приращение функционала окажется отрицательным и удаление этого объекта будет нецелесообразным.

Таким образом необходимо определить *окрестность значащих соседей*, т.е. найти такое число соседей, которое будет достаточно для однозначного определения целесообразности исключения конкретного объекта из множества прецедентов.

3.2 Определение числа значащих соседей

Цель: для каждого объекта x_e из исходной обучающей выборки по значению первых компонент вектора $\delta(e)$ разности исходного комбинаторного функционала и функционала, вычисленного для выборки, у которой объект x_e исключен из обучения, определить s-окрестность ближайших соседей, по результатам которых будет дан однозначный ответ о признании объекта x_e шумовым и исключении его из обучающей выборки. Т.о. необходимо найти такое число s, что на основании значения компонент $\delta_i(e),\ i=1,\ldots s$ будет однозначно определен знак приращения функционала ΔQ_c.

Для определения шумовых объектов:

$$\begin{cases} \Delta Q_c(x_e) \geq 0 \Rightarrow x_e - \textit{шумовой объект}, \\ \Delta Q_c(x_e) < 0 \Rightarrow x_e - \textit{опорный объект}. \end{cases} \quad (2)$$

Для определения периферийных объектов, приращение функционала можно сравнивать не с 0, а некоторым малым значением ε :

$$\begin{cases} -\varepsilon \leq \Delta Q_c(x_e) < 0 \Rightarrow x_e - \textit{периферийный объект}, \\ \Delta Q_c(x_e) < -\varepsilon \Rightarrow x_e - \textit{опорный объект}. \end{cases} \quad (3)$$

Пусть теперь по первым компонентам приращения функционала сделан вывод о его знаке. Необходимо оценить насколько максимально значение оставшихся компонент может повлиять на значение приращения.

Обозначим $\Delta^s Q_c(x_e) = \sum_{m=1}^{s} \delta_m(e)\widetilde{C}(m)$. Пусть $\Delta^s Q_c(x_e) > 0$, т.е. по первым *s*-компонентам, объект x_e признан шумовым.

Рассмотрим процедуру вычисления приращения функционала $\Delta Q_c(x_e)$. Это соответствует покомпонентному вычислению приращения вектора профиля компактности исходной выборки и выборки с исключенным из обучения объектом x_e, т.е. вычислению

$$\delta_m(e) = \frac{1}{L}\sum_{i=1}^{L} q_i^m(x_e) \stackrel{\Delta}{=} \frac{1}{L}\sum I(x_i, y^*(x_{im})) - I(x_i, \widetilde{y}^*(x_{im})),$$

где $\widetilde{y}^*(x_{im})$ – классификация *m*-го соседа объекта x_i, при удаленном объекте x_e. Обозначим через $\gamma(x_i, x_e)$ – порядковый номер объекта x_e в последовательности упорядоченных по возрастанию расстояния до x_i оставшихся *L-1*объектов, т.о. x_e является для объекта x_i $\gamma(x_i, x_e)$-ым соседом. Исходя из введенных обозначений получаем:

$$q_i^m(x_e) = \begin{cases} 0, \textit{если } m < \gamma(x_i, x_e), \\ I(x_i, y^*(x_{im})) - I(x_i, y^*(x_{im+1})), \textit{если } m \geq \gamma(x_i, x_e). \end{cases}$$

Отсюда следуют следующие свойства приращений компонентов профиля компактности для одного объекта:

1) $q_i^m(x_e) = -1$, *если* $r_m(x_i) = 0 \,\&\, r_{m+1}(x_i) \,\&\, m \geq \gamma(x_i, x_e)$.

2) Если $q_i^m(x_e) = -1$, то $\exists n > m : (q_i^j(x_e) = 0, \forall j = \overline{m+1, n-1}) \,\&\, (q_i^n(x_e) = 1)$

или $q_i^j(x_e) = 0, \forall j = \overline{m+1, k}$.

3) Если $q_i^m(x_e) = 1$, то $\exists n > m : (q_i^j(x_e) = 0, \forall j = \overline{m+1, n-1}) \,\&\, (q_i^n(x_e) = -1)$

или $q_i^j(x_e) = 0, \forall j = \overline{m+1, k}$.

Оценим сумму оставшихся компонент снизу, что приводит к уменьшению приращения. $\sum_{m=s+1}^{k}\delta_m(e)\widetilde{C}(m)\geq-\frac{1}{2}\widetilde{C}(s+1)$. Минимальное значение суммы оставшихся компонент, соответствует случаю, когда объект x_e был s-ым соседом для всех «своих» объектов, а $s+1$-ым соседом для них был уже объект из другого класса, т.е. $q_i^{s+1}(x_e)=-1$, при $\gamma(i,e)\leq s+1\,\&\,r_{s+1}(x_i)=0\,\&\,r_{s+2}(x_i)=1$. В этом случае по 2)-му свойству $q_i^m(x_e)$ следует, что

$$\sum_{m=s+1}^{k}q_i^m(x_e)\widetilde{C}(m)\geq-\widetilde{C}(s+1), \qquad (**)$$

т.к. если $\exists n: q_i^n(x_e)=1$, то $\sum_{m=s+1}^{k}q_i^m(x_e)\widetilde{C}(m)\geq-\widetilde{C}(s+1)+\widetilde{C}(n)>-\widetilde{C}(s+1)$.

Для «чужих» объектов значение профиля компактности для $s+1$-ых соседей не изменилось при удалении объекта x_e, т.е. $q_i^m=0$, *для* $\forall m\geq s+1$. Т.к. в противном случае значение приращения функционала только увеличивается, действительно, если $s+1\leq\gamma(i,e)\leq k\,\&\,r_{\gamma(i,e)}(x_i)=1\,\&\,\exists n>\gamma(i,e): r_n(x_i)=0$, то $q_i^{n-1}(x_e)=1$ *и* $\sum_{m=s+1}^{k}q_i^m(x_e)\widetilde{C}(m)>0$.

Предположим, что число объектов в обоих классах равны $\frac{L}{2}$, тогда, производя суммирование по всем объектам исходной выборки и, применяя неравенство (**), получим, что

$$\sum_{m=s+1}^{k}\delta_m(e)\widetilde{C}(m)=\sum_{m=s+1}^{k}\frac{1}{L}\left(\sum_{i=1}^{L}q_i^m(x_e)\right)\widetilde{C}(m)=\frac{1}{L}\sum_{i=1}^{L}\sum_{m=s+1}^{k}q_i^m(x_e)\widetilde{C}(m)\geq\frac{1}{L}\frac{L}{2}\left(-\widetilde{C}(s+1)\right)=-\frac{1}{2}\widetilde{C}(s+1)$$

Аналогично, с незначительными изменениями, можно получить оценку сверху, т.е.

$$\sum_{m=s+1}^{k}\delta_m(e)\widetilde{C}(m)\leq\frac{1}{2}\widetilde{C}(s+1).$$

Таким образом исходное приращение функционала качества удовлетворяет следующим неравенствам:

$$\Delta^s Q_c(x_e)-\frac{1}{2}\widetilde{C}(s+1)\leq\Delta Q_c(x_e)=\Delta^s Q_c(x_e)+\sum_{m=s+1}^{k}\delta_m(e)\widetilde{C}(m)\leq\Delta^s Q_c(x_e)+\frac{1}{2}\widetilde{C}(s+1). \quad (4)$$

Т.к. комбинаторные множители $\widetilde{C}(m)$ быстро убывают с ростом m, то можно предположить, что для однозначного определения целесообразности удаления объекта x_e будет достаточно исследовать s первых компонент вектора приращения профиля компактности. Найдем отношение предыдущего комбинаторного множителя к последующему:

$$\frac{\widetilde{C}(m)}{\widetilde{C}(m+1)}=\frac{C_{L-m-1}^{l-1}}{C_{L-m-2}^{l-1}}=\frac{(L-m-1)!(l-1)!(L-m-l-1)!}{(l-1)!(L-m-l)!(L-m-2)!}=\frac{L-m-1}{L-m-l},$$

$$\theta \overset{\Delta}{=} \frac{L}{k} < \frac{L-m-1}{L-m-l} \le \frac{L-s-1}{k-S} \overset{\Delta}{=} \theta', \ \forall m = \overline{1,s}.$$

Используя введенные обозначения, получим следующие неравенства, разделив оба неравенства (4) на $\widetilde{C}(s+1)$:

$$\sum_{m=1}^{s} \delta_m(e)\theta^{s+1-m} - \frac{1}{2} < \frac{1}{\widetilde{C}(s+1)} \Delta^s Q_c(x_e) + \sum_{m=s+1}^{k} \delta_m(e)\widetilde{C}(m) \le \sum_{m=1}^{s} \delta_m(e)\theta'^{s+1-m} + \frac{1}{2}. \quad (5)$$

Т.к. деление на $\widetilde{C}(s+1)$ не меняет знак приращения функционала, то на основании (5) сделать вывод об исключении объекта x_e можно, применяя следующее правило:

1) если $\sum_{m=1}^{s} \delta_m(e)\theta^{s+1-m} \ge \frac{1}{2}$, то $\Delta Q_c(x_e) > \left(\sum_{m=1}^{s} \delta_m(e)\theta^{s+1-m} - \frac{1}{2} \right) \widetilde{C}(s+1) > 0$ и из правила (2) следует, что x_e – шумовой объект.

2) если $\sum_{m=1}^{s} \delta_m(e)\theta'^{s+1-m} \le -\frac{1}{2}$, то $\Delta Q_c(x_e) \le \left(\sum_{m=1}^{s} \delta_m(e)\theta'^{s+1-m} + \frac{1}{2} \right) \widetilde{C}(s+1) \le 0$ и из правила (2) следует, что x_e – опорный объект.

Остается составить алгоритм определения параметра s – числа значащих соседей для объекта x_e:

1) s=1, вычисляем $\delta_1(e)$, тогда

$$\begin{cases} \left.\begin{array}{l} \delta_1(e)\theta \ge \frac{1}{2}, \ x_e - \textit{шумовой объект} \\ \delta_1(e)\theta' \le -\frac{1}{2}, \ x_e - \textit{опорный объект} \end{array}\right\} \textit{переходим к шагу 3)} \\ \textit{иначе переходим к шагу 2)} \end{cases},$$

2) s=s+1, вычисляем $\Theta^s \overset{\Delta}{=} \sum_{m=1}^{s} \delta_m(e)\theta^{s+1-m}$ и $\Theta'^s \overset{\Delta}{=} \sum_{m=1}^{s} \delta_m(e)\theta'^{s+1-m}$, тогда

$$\begin{cases} \left.\begin{array}{l} \Theta^s \ge \frac{1}{2}, \ x_e - \textit{шумовой объект} \\ \Theta'^s \le -\frac{1}{2}, \ x_e - \textit{опорный объект} \end{array}\right\} \textit{переходим к шагу 3)} \\ \textit{иначе возвращаемся к шагу 2)} \end{cases}$$

3) Завершаем вычисление остальных компонент вектора приращения, s – найденное число значащих соседей для объекта x_e.

Аналогично проводятся рассуждения относительно исключения непредставительных объектов, стой лишь разницей, что на основе правила (3) приращение функционала сравнивается не с 0, а с некоторым малым значением ε. Тогда неравенства в предыдущем правиле заменятся на следующие:

1) $\sum_{m=1}^{s} \delta_m(e)\theta^{s+1-m} \geq \frac{1}{2} - \frac{\varepsilon}{\widetilde{C}(s+1)}$,

2) $\sum_{m=1}^{s} \delta_m(e)\theta'^{s+1-m} \leq -\frac{1}{2} + \frac{\varepsilon}{\widetilde{C}(s+1)}$.

3.3 Критерий останова при исключении периферийных объектов

Точного критерия остановки отброса периферийных объектов, т.е. критерия перехода периферийных объектов в опорные, нет. Поэтому одним из возможных способов определения такого критерия является введение некоторого малого порога $\varepsilon > 0$. С его помощью процесс перехода периферийных объектов в опорные и останова отброса объектов определяется следующим образом:

$$\begin{cases} -\varepsilon \leq \Delta Q_c(x_e) < 0 \Rightarrow x_e - \textit{периферийный объект}, \\ \Delta Q_c(x_e) < -\varepsilon \Rightarrow x_e - \textit{опорный объект}. \end{cases}$$

Этот порог может настраиваться в процессе обучения, а также выбираться на основе заранее известной, априорной информации. Также данный критерий может устанавливаться непосредственно с помощью эксперта, который производит разбиение объектов исходной обучающей выборки на три группы: шумовые, периферийные и опорные, с помощью графика зависимости значения функционала $Q_c(\mu, X^L)$ от числа исключенных объектов (см. рис. 2).

3.4 Последовательность исключения шумовых объектов

При исключении шумовых объектов из обучающей выборки возможен более точный алгоритм этого отбора, определяющий однозначное *последовательное* исключение объектов с максимальным значением приращения функционала. Этот *алгоритм последовательного исключения шумовых объектов* заключается в том, что после сортировки векторов приращения *δ(e),* из множества прецедентов исключается один шумовой объект, с максимальным значением этого вектора. После этого происходит пересчет всех векторов *δ(e)* без учета исключенного объекта. Снова выбирается объект с максимальны значением приращения функционала и исключается из обучающей выборки. Такой процесс повторяется до тех пор, пока максимальный вектор приращения окажется отрицательным. В этом алгоритме происходит пересчет векторов приращения, после удаления каждого объекта, в результате чего производится огромный объем вычислений, и практическое использование такого алгоритма ставится под вопросом уже при небольших размерах исходных данных, порядка 1000 объектов. Вместе с тем практические исследования результатов работы

алгоритма последовательного исключения объектов, показывают что в итоге множество отобранных им опорных объектов практически совпадает с множество прецедентов, полученных в результате работы основного алгоритма CCV, описанного выше, реализующий *групповое исключение шумовых объектов*. Это позволяет сделать вывод о нецелесообразности использования последовательного исключения в результате, которого мы получаем, по сравнению с основным алгоритмом, только правильный однозначный порядок исключения шумовых объектов, значительно проигрывая в скорости.

Таким образом наблюдается некоторая устойчивость шумовых объектов, которые остаются таковыми при исключении все группы объектов, признанных на данном этапе шумовыми. В итоге можно значительно сократить объем вычислений, исключая сразу всю группу шумовых объектов, а не последовательно удаляя «максимально» шумовой объект. Данное предположение подтверждается на экспериментах.

4 Вычислительные эксперименты

4.1 Изменение функционала CCV в процессе отсева объектов

В предложенном в данной работе алгоритме CCV примечательным является тот факт, что для исключения как шумовых, так и периферийных объектов, применяется один и тот же функционал полного скользящего контроля. На приведенном ниже рисунке показан график изменения значения функционала полного скользящего контроля, при последовательном исключении объектов, отсортированных по максимуму приращения функционала качества ΔQ_c (обозначен синим цветом).

Данный график показывает, что процесс исключения объектов проходит три стадии. Сначала удаляются шумовые объекты — значение функционала уменьшается. Затем удаляются неинформативные периферийные объекты — значение функционала не изменяется, затем начинает несущественно увеличиваться. Наконец, остаются только опорные объекты — их удаление приводит к заметному увеличению функционала.

Опорные объекты находятся «в среднем слое» не слишком граничных и не слишком периферийных объектов. Граничные объекты часто оказываются шумовыми выбросами, а периферийные — просто бесполезны, если они находятся в плотном окружении объектов своего класса. Чем меньше отношение шум/сигнал, тем ближе средний слой может подходить к границе классов. Отметим, что идея среднего слоя близка к идее машины релевантных векторов.

Второй график отображает зависимость доли ошибок на тестовой выборке от числа исключенных из обучения объектов (обозначен красным цветом). Тем самым, он характеризует обобщающую способность описанного алгоритма. При сравнении этого графика с первым, оказывается, что его минимум достигается на том же самом опорном множестве объектов, на котором достигается минимум функционала полного скользящего контроля. Данный факт означает, что в результате отбора опорного множества объектов, на основе оптимизации функционала CCV, не происходит переобучения данного алгоритма, вследствие одинакового поведения частоты ошибок классификации в процессе исключения объектов на обучающей и тестовой выборке.

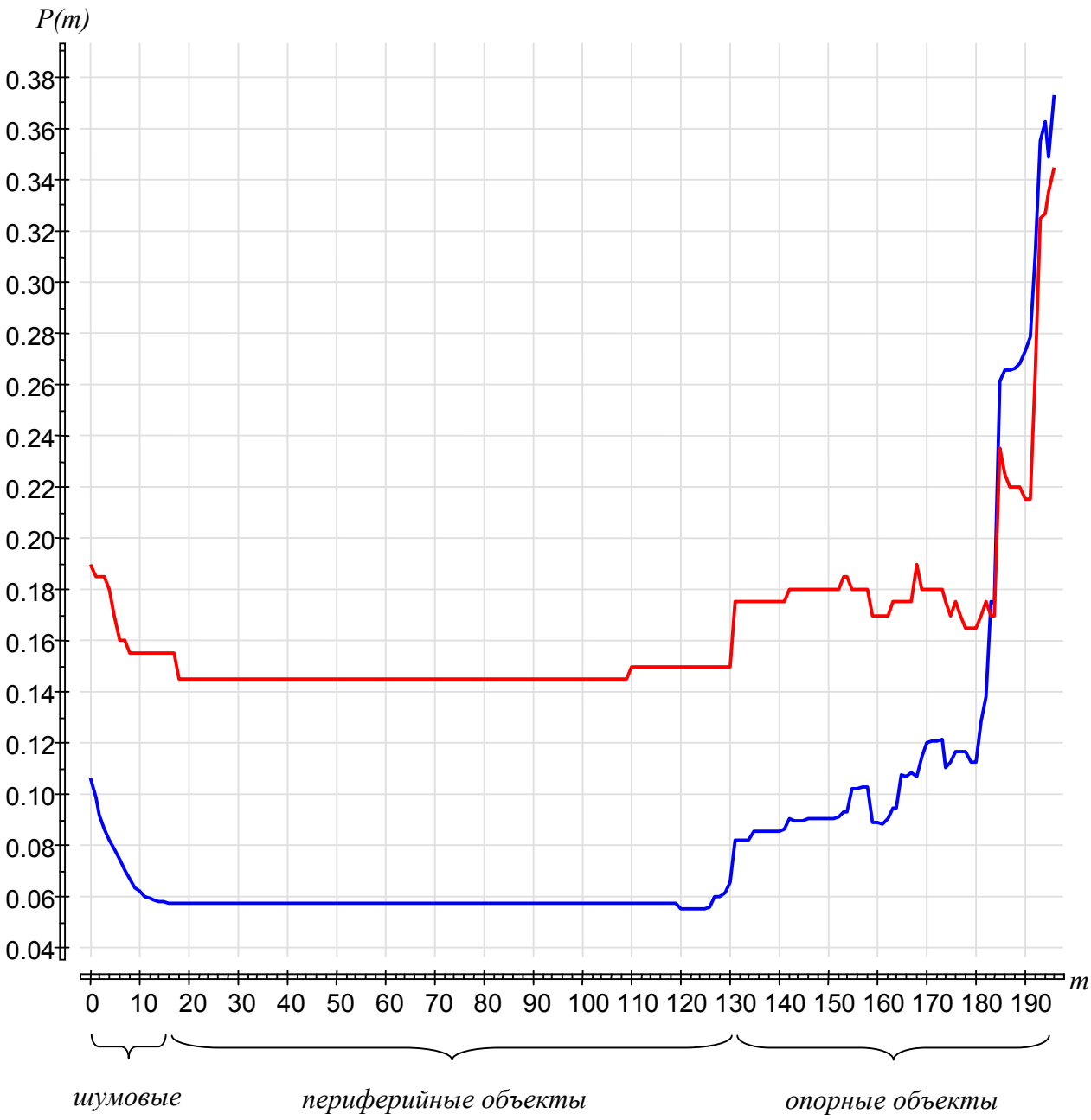

Рис. 2: Синяя линия: график зависимости значения функционала $Q_c(\mu, X^L)$ от числа исключенных объектов; красная линия: график зависимости доли ошибок на тестовой выборке от числа исключенных из обучения объектов.

На следующем рисунке 3 показано, как с помощью описанного выше алгоритма можно исходную обучающую выборку разделить на три большие группы объектов: шумовые, периферийные и опорные.

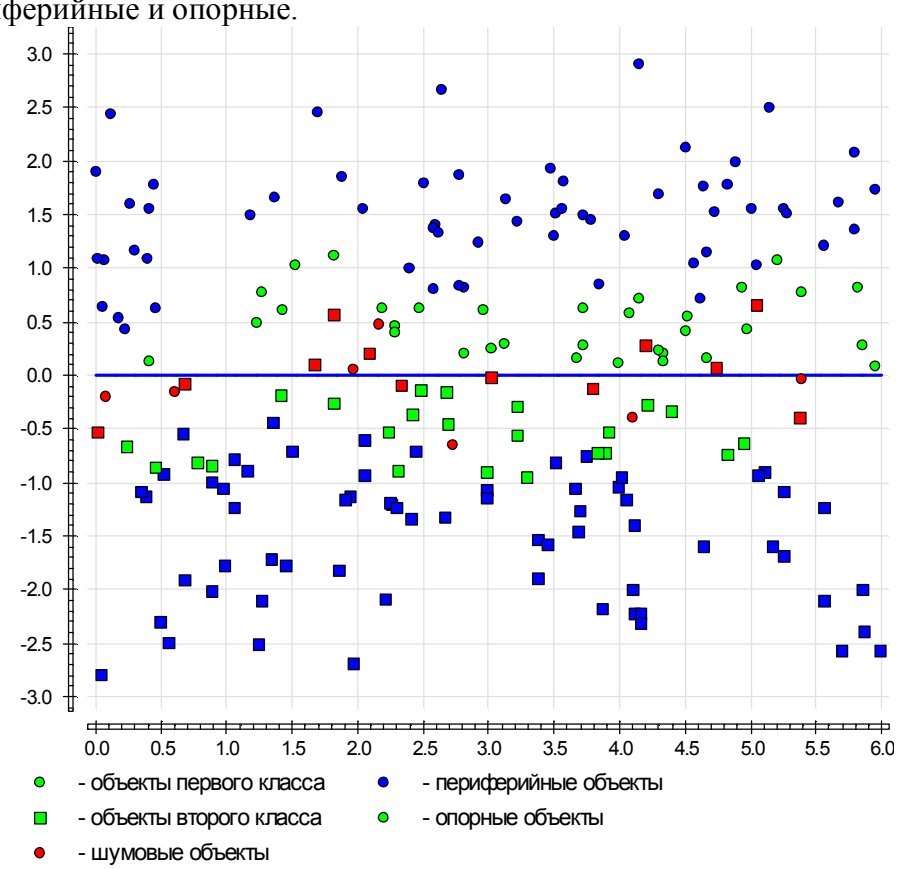

Рис. 3: Разделение исходной обучающей выборки на три группы: шумовые, опорные и периферийные объекты (на рисунке группы обозначаются цветом).

Для сравнения последовательного и группового исключения объектов, а также наглядного представления улучшения профиля компактности исходной выборки алгоритмом CCV, был построен график, на котором изображены профиль компактности исходной обучающей выборки, выборки, полученной после отсева шумовых и периферийных объектов последовательным и групповым алгоритмом CCV, а также выборки со случайно исключенными из нее объектами. На данном графике видно, что алгоритм CCV, как последовательный так и групповой вариант, значительно уменьшают профиль компактности на начальном ее отрезке при малых m, что оказывается достаточным для уменьшения всего значения функционала полного скользящего контроля $Q_c(\mu, X^L)$ по сравнению с исходной выборкой.

Вместе с тем групповой и последовательный вариант алгоритма показывают практический одинаковый результат при начальных значения профиля компактности, это сви-

детельствует о нецелесообразности применения последовательного исключения шумовых объектов для улучшения профиля компактности, которое значительно превышает по объемам вычислений групповое исключение. Также на графике видно, что простое случайное исключение объектов не приносит никакого положительного эффекта, а, наоборот, сильно ухудшает профиль компактности по сравнению с исходной выборкой.

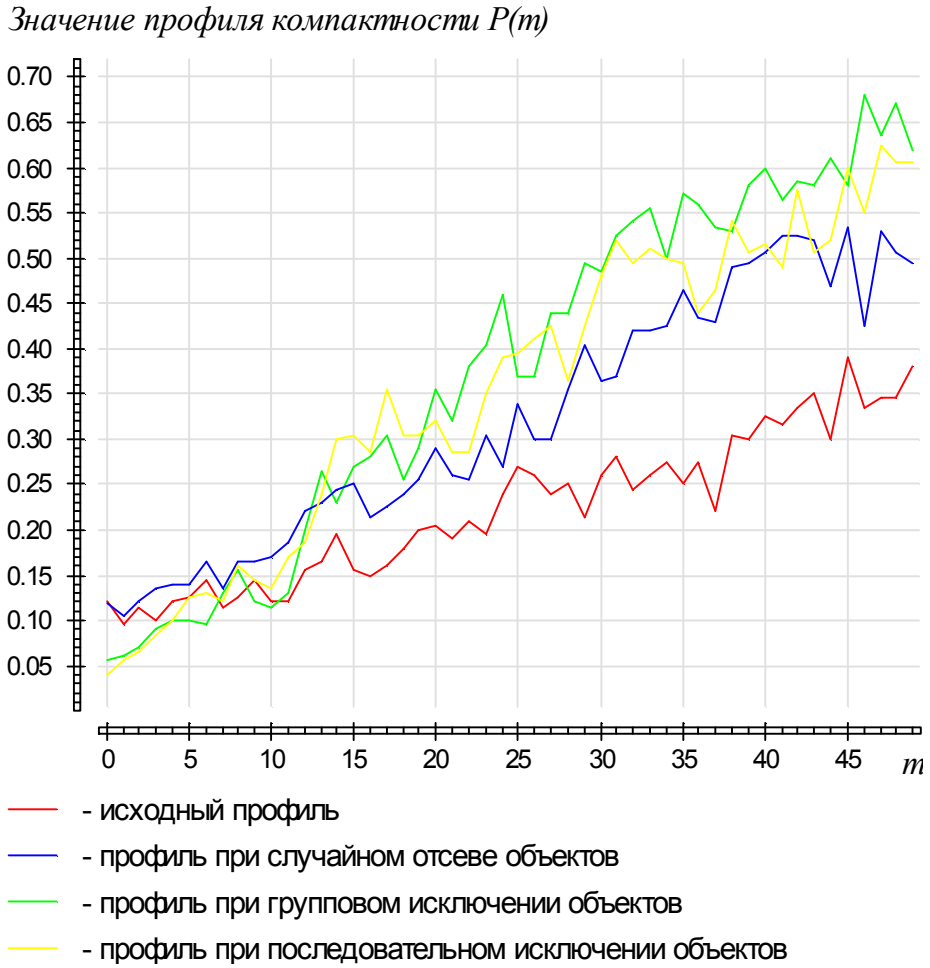

Рис. 4 Улучшение профиля компактности после отсева шумовых объектов

4.2 Сравнение алгоритмов на модельных данных

Для проверки описанных теоретических результатов, тестирования предложенного в данной работе алгоритма и его сравнения с другими существующими алгоритмами выбора опорных объектов были проведены различные эксперименты на модельных данных. Модельные данные представляют собой два класса объектов. Несмотря на то, что алгоритм имеет дело только с попарными расстояниями между объектами, для генерации модельных данных необходимо сгенерировать объекты с признаковым описанием. Для наглядности и простоты было выбрано двухмерной признаковое пространство. В этом пространстве классы представляют собой прямоугольные области. Область первого класса определяется четырьмя точками: (0,0), (6,0), (6,3) и (0,3), аналогично область второго

класса: (0,-3), (6,-3), (6,0) и (0,0). Таким образом граница класса проходит по оси *Ox* (на рисунке обозначена жирной синей линией). Объекты внутри границы классов распределены равномерно. Также к этим объектам добавлены шумовые объекты, которые имеют нормальное распределение по оси *Oy* с параметрами (0, 1). Число объектов в обучающей и тестовой выборке равно 200. На следующем рисунке показан результат отбора опорных объектов описанным выше алгоритмом CCV. Изменением цвета показана последовательность отсева объектов из обучения: чем светлее цвет объекта, тем раньше он был исключен.

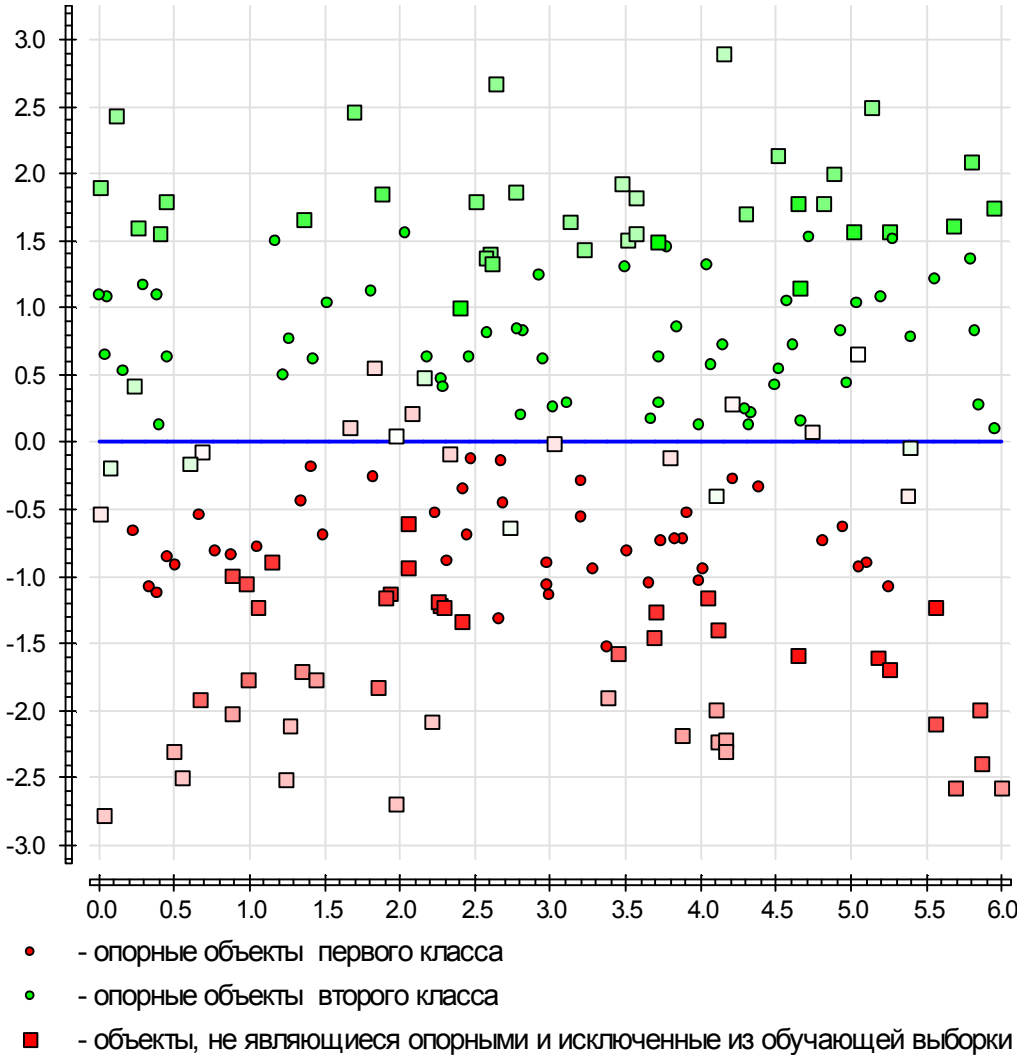

Рис. 5 Опорное множество объектов, полученное алгоритмом CCV

На этом графике видно, что алгоритм отбросил все шумовые объекты, добавленные в исходную обучающую выборку, также удалены далекие от границы, периферийные объекты. Оставшееся после выполнения алгоритма множество объектов «среднего слоя» и является опорным. Для проверки качества алгоритма вычислим частоту ошибок на независимо сгенерированной тестовой выборке, имеющее тоже распределение, что и обучаю-

щая. На рис. 6 изображена тестовая выборка и отмечены объекты, ошибочно классифицированные алгоритмом CCV. В итоге частота ошибки оказалась равной 0.145.

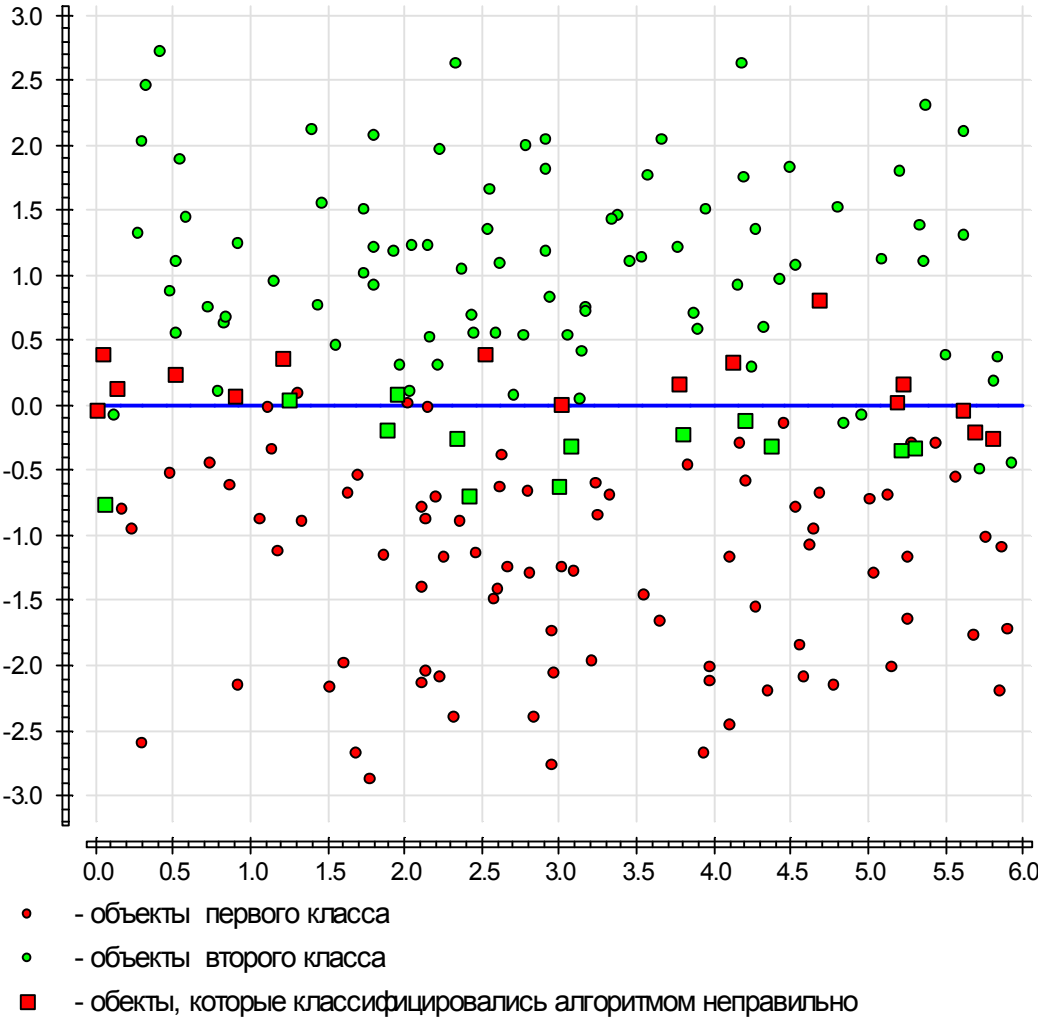

Рис.6 Результат работы алгоритма CCV над тестовыми данными (Частота ошибки равна 0.145)

Для сравнения предложенного в этой работе алгоритма с другими метрическими алгоритмами, применим алгоритм λ-СТОЛП к тем же самым исходным данным: обучающей и тестовой выборки. Опорное множество объектов, полученное в результате обучения алгоритма λ-СТОЛП, изображено на рис. 7. Результат работы этого алгоритма на аналогичной тестовой выборке показан на рис. 8, где частота ошибок получилась равной 0.205, что значительно больше, чем у алгоритма CCV. В основном это объясняется тем, что все шумовые объекты, отставлены алгоритмом λ-СТОЛП в результате обучения сохраняет в опорном множестве все шумовые объекты. Таким образом описанный в данной работе алгоритм отбора опорных объектов CCV показывает лучшее качество работы по сравнению с эмпирическими алгоритмам на обучающих выборках содержащих значительное количество шумовых объектов. Следует также отметить, что применение алго-

ритма CCV на тестовых выборках, не содержащих шумовых объектов, приводит к практически нулевому количеству ошибок.

Аналогичные результаты были получены на других модельных данных, в которых граница разделения двух классов проходи по синусоиде. Частота ошибок на тестовой выборке для алгоритма CCV оказалась равной 0.025, а для алгоритма λ-СТОЛП равна 0.16 .

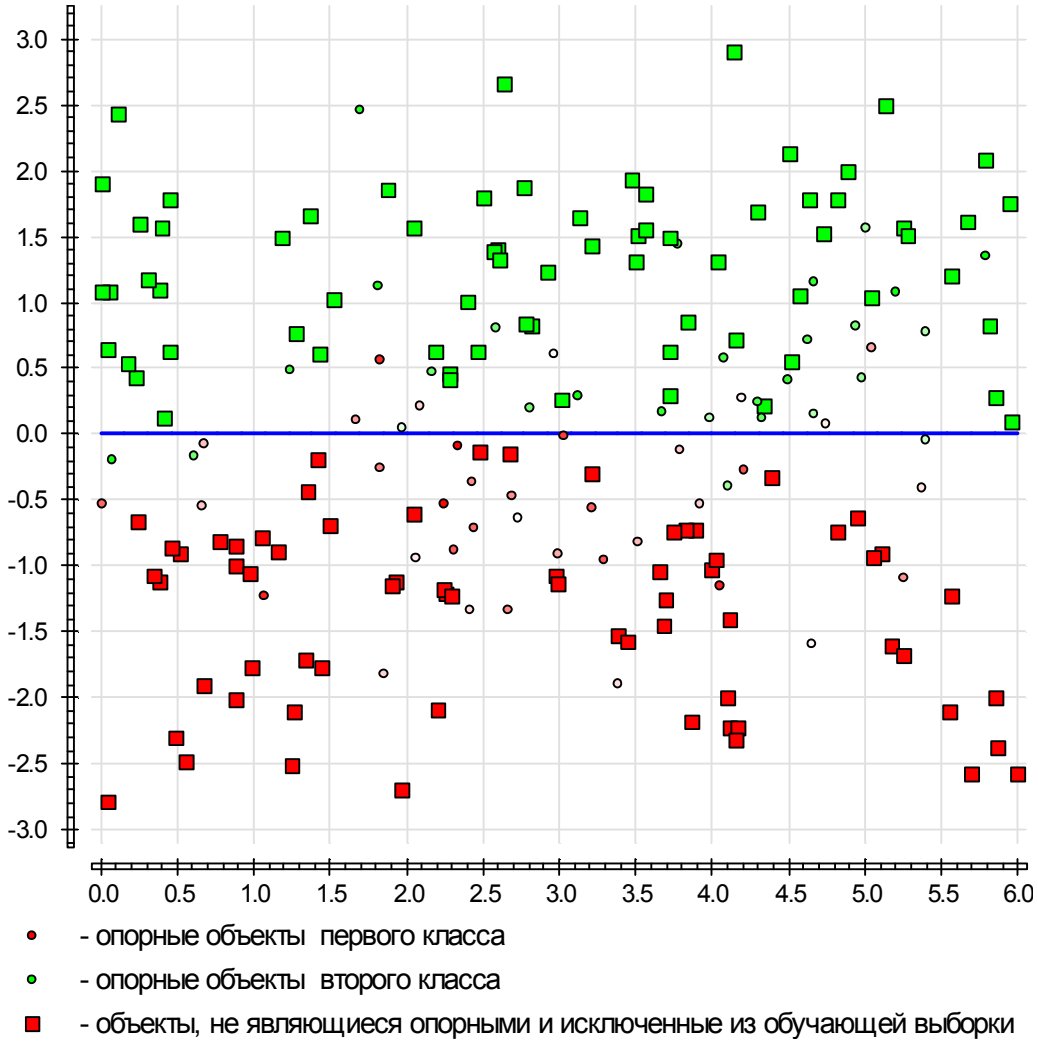

Рис. 7 Опорное множество объектов, полученное алгоритмом λ-STOLP

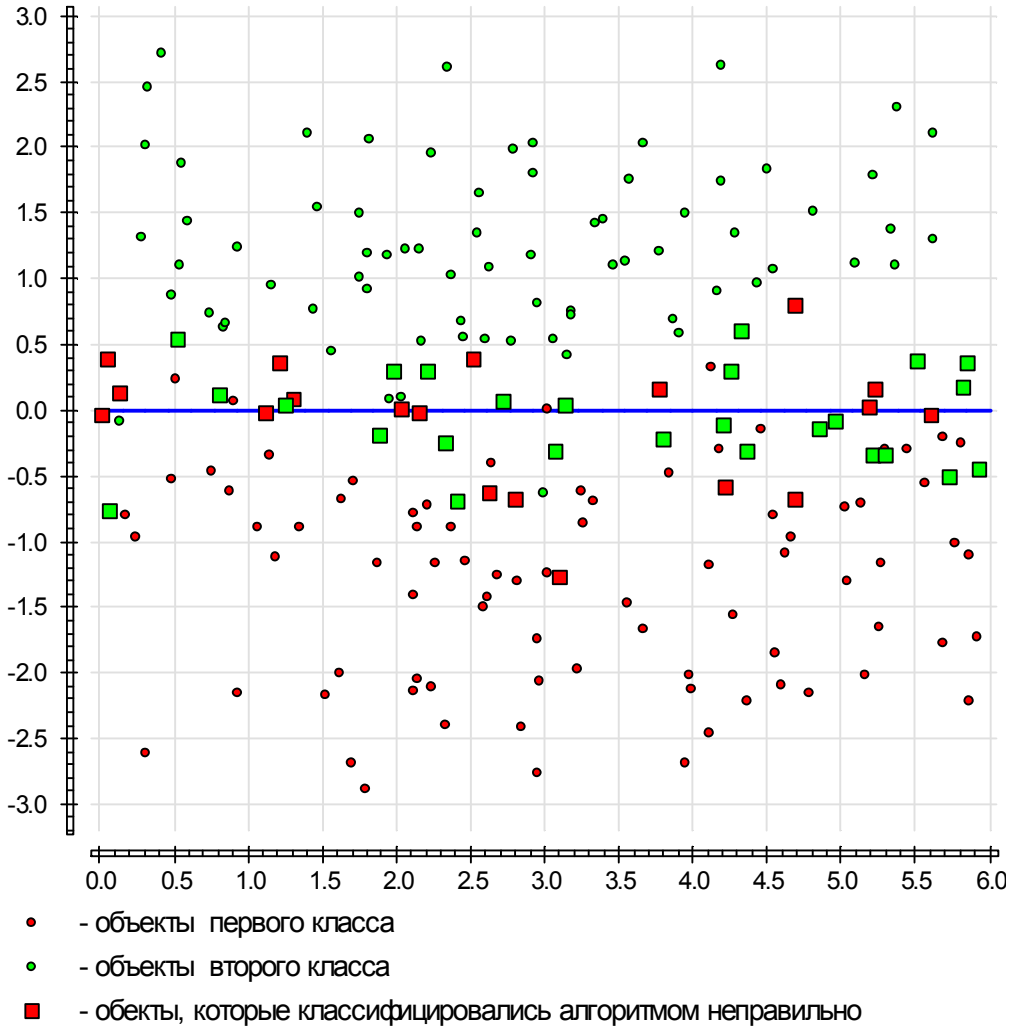

Рис.8 Результат работы алгоритма STOLP на тестовых данных.

(Частота ошибки равна 0.205)

5 Выводы

Данная работа является частью исследований, направленных на разработку алгоритмов классификации, позволяющих оценивать обобщающую способность алгоритма в процессе его синтезе по эмпирическим данным и управлять этим процессом.

Основные результаты, полученные в работе:

- Разработана процедура отбора опорных объектов для метрических алгоритмов классификации, позволяющая повысить качество классификации, сократить время классификации и объем хранимых данных.
- Экспериментально показано, что использование функционала полного скользящего контроля в качестве отбора опорных объектов позволяет строить алгоритмы более высокого качества, чем другие эвристические критерии.

Список литературы

[1] *Айзерман М. А., Браверманн Э. М., Розоноэр Л. И.* Метод потенциальных функций в теории обучения машин. — М.: Наука, 1970. — Р. 320.

[2] *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Найка, 1979.

[3] *Воронцов К. В.* Комбинаторные оценки качества обучения по прецедентам // Докл. РАН. — 2004. — Т. 394, № 2. — С. 175 – 178.

[4] *Воронцов К. В.* Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики. — 2004. — Т. (в печати), № ? — С. ?

[5] *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999.

[6] *Mullin M., Sukthankar R.* Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning. — 2000. http://citeseer.ist.psu.edu/309025.html.