DOI: 10.18721/JCSTCS.12105
УДК 519.712.3, 004.896:004.048

РОБАСТНЫЕ АЛГОРИТМЫ КЛАССИФИКАЦИИ ДАННЫХ, ПОЛУЧЕННЫЕ ГРУППОЙ РОБОТОВ, С ИСПОЛЬЗОВАНИЕМ МНОЖЕСТВ ВЕСОВ

С.Г. Попов, Л.В. Уткин, В.С. Заборовский

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Предложено три адаптивных робастных алгоритма обучения системы группы роботов при условии, что каждое наблюдение, полученное роботами, является многозначным, состоящим из нескольких элементов. Причина многозначных данных заключается в том, что роботы в системе предоставляют различные измерения в качестве одного наблюдения или в один момент времени. В основе алгоритмов – множества весов или интервальные веса определенного вида для всех элементов обучающего множества. Кроме того, для формализации многозначных данных и модификации весов в процессе получения новых данных рекомендовано использование интервальной модели Дирихле. Первый алгоритм – это модификация метода опорных векторов, учитывающая многозначные данные. Второй алгоритм – модификация алгоритма AdaBoost для многозначных данных. Третий алгоритм – комбинация AdaBoost и интервальной модели Дирихле. Все алгоритмы являются робастными и используют минимаксную стратегию принятия решений.

Ключевые слова: группа роботов, классификация, метод опорных векторов, многозначные наблюдения, модель Дирихле.

Ссылка при цитировании: Попов С.Г., Уткин Л.В., Заборовский В.С. Робастные алгоритмы классификации данных, полученные группой роботов, с использованием множеств весов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2019. Т. 12. № 1. С. 44–54. DOI: 10.18721/JCSTCS.12105.

ROBUST ALGORITHMS OF DATA CLASSIFICATION OBTAINED USING SET OF WEIGHTS BY GROUP OF ROBOTS

S.G. Popov, L.V. Utkin, V.S. Zaborovsky

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

Three adaptive robust learning algorithms for a group of robots are proposed in the paper, provided that each observation obtained by the robots consists of several elements, i.e, is multi-valued. The reason for the multi-valued data is that the robots in the system provide different measurements for a single external parameter observation at a time. The algorithms are based on sets of weights or interval weights of a certain type for all elements of the training set. In addition, to formalize multivalued data and modify weights in the process of obtaining new data, it is proposed to use the Dirichlet interval model. The first algorithm is a modification of the support vector machine that takes into account multivalued data. The second algorithm is a modification of the AdaBoost algorithm for multi-valued data. The third algorithm is a combination of AdaBoost and the Dirichlet interval model. All algorithms are robust and use a minimax decision-making strategy.

Keywords: group of robots, classification, support vector method, multi-valued observations, Dirichlet model.

Citation: Popov S.G., Utkin L.V., Zaborovsky V.S. Robust algorithms of data classification obtained using set of weights by group of robots. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2019, Vol. 12, No. 1, Pp. 44–54. DOI: 10.18721/JCSTCS.12105.

Введение

К системам, состоящим из группы роботов (СГР), в последнее время возрастает интерес благодаря ряду факторов, включая способность СГР решать сложные задачи более эффективно по сравнению с одним роботом [6, 10]. В СГР роботы обычно оборудованы рядом сенсоров, использующихся для получения информации о внешней среде. В работе [7] отмечено, что каждый сенсор может получать различные значения характеристик среды, однако альтернативная интерпретация информации, полученнной одним и тем же сенсором, также может быть весьма полезна. Одной из важных проблем обучения СГР является эффективное комбинирование выходов множества распределенных сенсоров, которые могут включать устройства GPS, датчики температуры, высотомеры, системы наблюдения и т. д. [1].

Существует большое число подходов к комбинированию информации в процессе обучения СГР (см., например, [3, 5, 8, 14]), в основном использующих различные весовые схемы (в терминах вероятностей или других характеристик) для того, чтобы различать источники обучающих данных и комбинировать их в соответствии с определенными правилами, учитывающими качество или надежность данных, полученных от разных сенсоров. Однако большинство подходов предполагают, что имеется большое множество обучающих примеров для назначения весов сенсорам. Это предположение может нарушаться во многих случаях особенно на ранней стадии обучения, когда сложно оценить каждого робота или его сенсоры, чтобы использовать имеющиеся весовые схемы.

Можно выделить две стратегии обучения СГР. В соответствии с первой стратегией выполняется совместная классификация роботов на основе одного метаклассификатора, обученного на всех данных, полученых от сенсоров и обратной связи, обеспечивамой «учителем». Эта стратегия полезна на начальном этапе обучения, когда неизвестно, как разные роботы ведут себя в своей группе и насколько надежна информация от датчиков, предоставляемая каждым роботом. Вторая стратегия заключается в том, что каждый робот обучает свой собственный классификатор [2], используя признаки, извлеченные из набора локально доступных размеченных примеров, соответствующих информации, полученной с сенсоров конкретного робота и других роботов.

В настоящей статье описывается только первая стратегия. Мы изучаем случай, когда трудно или просто невозможно присвоить веса отдельным датчикам, чтобы объединить их выходы, используя весовые схемы. Основная сложность совместного использования обучающих данных от нескольких датчиков на начальном этапе обучения заключается в том, что мы не можем брать данные каждого датчика в качестве отдельных обучающих примеров. Возьмем, например, датчики температуры, которые в определенный момент времени измеряют температуру окружающей среды для группы роботов. Каждый датчик предоставляет информацию о температуре одного и того же объекта примерно в одно и то же время. Следовательно, набор измерений температуры в этом случае следует рассматривать как единый многозначный обучающий пример. Конечно, мы можем использовать, например, некоторую среднюю температуру для обучения. Однако это правило комбинирования говорит о том, что все роботы одинаково надежны и точны. Такое предположение является слишком сильным для того, чтобы быть действительным во многих приложениях. Начальный этап обучения характеризуется отсутствием соответствующих знаний. Поэтому мы предлагаем алгоритм обучения, учитывающий перечисленные выше особенности. Следует отметить, что в некоторых приложениях может быть доступна только начальная фаза обучения, и алгоритм обучения, предложенный для этой фазы, полностью используется в обучении СГР.

Одним из наиболее эффективных и популярных классификационных методов обучения СГР является метод опорных векторов – SVM. Другим эффективным методом является AdaBoost, описанный в [4]. Поэтому мы предлагаем модификацию AdaBoost с SVM особой формы в качестве слабого классификатора, которая учитывает тот факт, что обучающие данные СГР получены от набора неизвестных роботов или их датчиков. Кроме того, мы также модифицируем SVM так, что неточные сведения о роботах могут быть инкорпорированы в SVM для улучшения классификации СГР. Основная идея, лежащая в основе предлагаемых модификаций, заключается в следующем. Данные обучения от каждого робота рассматриваются в качестве обучающих примеров, но их веса заменяются некоторыми множествами весов, и каждый обучающий пример может рассматриваться как многозначные данные. Затем мы применяем робастную минимаксную стратегию, чтобы найти оптимальную решающую функцию, разделяющую многозначные наблюдения различных классов. Множества весов получаются из неточной доступной информации о роботах. Кроме того, мы предлагаем двойной адаптивный алгоритм. Первая адаптация выполняется AdaBoost путем изменения весов наблюдений. Вторая адаптация – обновление множеств весов наблюдений с многозначными значениями в соответствии с количеством правильно классифицированных измерений от каждого робота на каждой итерации AdaBoost.

Вторая адаптация реализована с помощью интервальной модели Дирихле [13]. Фактически, мы предлагаем три алгоритма обучения СГР. Самый простой из них – модификация SVM для учета многозначных данных. Второй алгоритм – AdaBoost с модифицированным SVM при многозначных данных. Третий алгоритм – модификация AdaBoost с модификацией интервальных весов роботов. Сложность алгоритмов не отличается от сложности соответствующих стандартных алгоритмов SVM и AdaBoost.

Формальная постановка задачи и SVM

Предположим, что у нас есть наблюдения или измерения от всех датчиков T роботов в каждый момент времени k. После момента времени n получаем обучающее множество $S = \{(\mathbf{A}_1, y_1), ..., (\mathbf{A}_n, y_n)\}$, где $\mathbf{A}_k$ – матрица, имеющая T строк $\mathbf{x}_1^{(k)}, ..., \mathbf{x}_T^{(k)}$, $k = 1, ..., m$, и m столбцов, таких что строка $\mathbf{x}_j^{(k)}$ – вектор всех измерений (признаков), полученных от j-го робота. Мы предполагаем, что имеется два класса (двоичная классификация) и $y_i \in \{-1, 1\}$. Цель обучения – построить классификатор c: $\mathbb{R}^{m \cdot T} \to \{-1, 1\}$, который максимизирует вероятность того, что $c(\mathbf{A}_i) = y_i$ для всех $i = 1, ..., n$.

Один из путей для классификации – поиск разделяющей функции $f(\mathbf{x}, \mathbf{w}, b)$, имеющей параметры $\mathbf{w}$, такие, что $\mathbf{w} = (w_1, ..., w_m) \in \mathbb{R}^m$ и $b \in \mathbb{R}$, например, $f(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \mathbf{x} \rangle + b$. Здесь $\langle \mathbf{w}, \mathbf{x} \rangle$ означает поточечное произведение векторов $\mathbf{w}$ и $\mathbf{x}$. Обозначим также $w = (\mathbf{w}, b)$. Предполагается, что после периода обучения каждый робот использует разделяющую функцию $f(\mathbf{x}, \mathbf{w}, b)$. Можно также использовать функцию $f(\mathbf{A}, \mathbf{w}, b)$, которая определяется для матрицы $\mathbf{A}$ в случае переноса обучения между роботами.

Один из простейших путей для решения задачи классификации – замена каждого столбца $\mathbf{A}_k$ числом, например, средним значением для всех элементов столбца, и применение стандартного SVM.

Для краткого описания известного SVM мы заменим множество S множеством $S^* = \{(\mathbf{x}_1^*, y_1), ..., (\mathbf{x}_n^*, y_n)\}$. Здесь $\mathbf{x}_i^*$ – вектор замененных значений для каждого признака. Пусть ϕ – отображение признаков $\mathbb{R}^m \to G$ такое, что точки данных отображаются в пространство большей размерности G. Другими словами, это отображение в G такое, что существует образ ϕ, который вычисляется при помощи простого ядра $K(\mathbf{x}_i^*, \mathbf{x}_j^*) = (\phi(\mathbf{x}_i^*), \phi(\mathbf{x}_j^*))$, например, Гауссова ядра. SVM минимизирует эмпирический риск с учетом сглаживающего или регуляризационного слагаемого $\langle \mathbf{w}, \mathbf{w} \rangle / 2$:

$$R(w) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C\sum_{i=1}^{n} l(y_i, \mathbf{x}_i^*, w). \quad (1)$$

Здесь C – настраиваемый «штрафной» параметр C, который устанавливает компромисс между эмпирическим риском и штрафным слагаемым [9]; $l(y_i, \mathbf{x}_i^*, w)$ – классификационная функция потерь. Так называемая петлевая функция потерь используется в SVM, то есть $l(y, \mathbf{x}, w) = \max(0, 1 - yf(w, \phi(\mathbf{x})))$. Отсюда классификатор SVM может быть представлен в виде следующей выпуклой задачи оптимизации со вспомогательными переменными ξ_i, $i = 1, ..., n$:

$$\min_{\xi, w} R(w) = \min_{\xi, w}\left(\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C\sum_{i=1}^{n} \xi_i\right), \quad (2)$$

при ограничениях

$$\xi_i \geq 0, \; y_i\left(\left\langle w, \phi(\mathbf{x}_i^*)\right\rangle + b\right) \geq 1 - \xi_i, \quad i = 1, ..., n. \quad (3)$$

Величина $C\xi_i$ – «штраф» за любые данные $\mathbf{x}_i^*$, которые либо лежат в полосе отступов на «корректной» стороне гиперплоскости ($\xi_i \leq 1$) либо на «некорректной» стороне гиперплоскости ($\xi_i > 1$).

Вместо минимизации прямой задачи оптимизации (2) с ограничениями (3), используем двойственную задачу:

$$\max_{\alpha}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i^*, \mathbf{x}_j^*)\right), \quad (4)$$

при ограничениях

$$\sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., n. \quad (5)$$

Здесь α_i, $i = 1, ..., n$ – множители Лагранжа или переменные оптимизации. После подстановки полученного решения в выражение для решающей функции f получаем двойственную решающую функцию:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}) + b. \quad (6)$$

Параметр b определяется с использованием опорных векторов $\mathbf{x}_i^*$ из следующего уравнения:

$$b = y_j - \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}_j^*). \quad (7)$$

SVM при многозначных обучающих данных

Представленный подход для работы с обучающим множеством S при замене его на S^* не может использоваться в режиме большого «шума» и при малой обучающей выборке, когда роботы предоставляют «скудные» измерения. Поэтому для создания процедуры робастной классификации предлагается другой подход для работы с обучающим множеством S.

Рассмотрим множество эмпирически ожидаемых значений риска $R(w)$ такое, что каждое значение из множества соответствует строке, скажем, $\mathbf{x}_j^{(k)}$ матрицы $\mathbf{A}_k$. Тогда существует верхняя граница для $R(w)$, которая определяется как

$$\overline{R}(w) = \max_{\mathbf{x}_i^{(k)} \in \mathbf{A}_k,\, k=1,\ldots,n} \sum_{i=1}^{n} l(y_i, \mathbf{x}_i^{(k)}, w). \quad (8)$$

Здесь ожидаемый риск максимизируется по всем $\mathbf{x}_i^{(k)}$ из $\mathbf{A}_k$, $k = 1, \ldots, n$. Верхняя граница $\overline{R}(w)$ соответствует робастной или пессимистической стратегии в том смысле, что выбирается «наихудший» элемент $\mathbf{x}_0^{(k)}$ из $\mathbf{A}_k$.

Предположим, что имеются строки $\mathbf{x}_0^{(k)} \in \mathbf{A}_k$ для $k = 1, \ldots, n$, которые обеспечивают максимальное значение ожидаемого риска. Тогда можно назначить ненулевые веса строкам, таким что веса других векторов $\mathbf{x}_i^{(k)} \neq \mathbf{x}_0^{(k)}$ равны нулю. Отсюда следует, что задача максимизации ожидаемого риска по строкам матриц $\mathbf{A}_1, \ldots, \mathbf{A}_n$ может быть преобразована в задачу максимизации ожидаемого риска по множеству весов. Это преобразование можно рассматривать как преобразование неопределенности, т. е. обучающие данные с неопределенностью измерений преобразуются в весовую или вероятностную неопределенность [12].

Поэтому мы расширяем обучающую выборку строками $\mathbf{x}_i^{(k)}$ такими, что расширенное обучающее множество имеет теперь $N = T \cdot n$ элементов, но эти элементы имеют различные веса. Обозначим вектор новых весов как $\pi = (\pi_1, \ldots, \pi_N)$. Введем также множество индексов $I_k = \{1 + (k-1)T, \ldots, T + (k-1)T\}$. О распределении π известно только то, что сумма весов всех строк из $\mathbf{A}_k$ равна $\sum_{i \in I_k} \pi_i = 1/n$, так как каждый элемент исходного обучающего множества имеет вес или векроятность $1/n$. Отюда следует, что множество $\mathcal{P}$, образованное всеми возможными распределениями π, является выпуклым, и существует верхняя граница для $R(w)$, которая записывается как

$$\overline{R}(w) = \max_{\pi \in \mathcal{P}} \sum_{k=1}^{n} \sum_{i \in I_k} \pi_i l(y_i, \mathbf{x}_i^{(k)}, w). \quad (9)$$

Важно отметить, что обучающее множество не просто расширяется. Добавляя новые элементы в обучающее множество, изменяются веса этих элементов. При этом о весах новых элементов известно только то, что они принадлежат множеству $\mathcal{P}$.

Теперь можно построить модификацию SVM с учитом робастной стратегии, которая формулируется в виде минимаксной задачи оптимизации:

$$\min_w \overline{R}(w) = \min_w \max_{\pi \in \mathcal{P}} R(w). \quad (10)$$

Зафиксируем переменные w и рассмотрим только задачу с переменными $\pi \in \mathcal{P}$ при фиксированных w. Верхняя граница для $R(w)$ может быть найдена решением следующей задачи оптимизации:

$$\overline{R}(w) = \max_{\pi \in \mathcal{P}} \sum_{k=1}^{n} \sum_{i \in I_k} \pi_i l(y_i, \mathbf{x}_i^{(k)}, w), \quad (11)$$

при ограничениях

$$\sum_{i \in I_k} \pi_i = \frac{1}{n},\ k = 1, \ldots, n,\quad \sum_{k=1}^{n} \sum_{i \in I_k} \pi_i = 1. \quad (12)$$

Ограничения получены из условия, что веса исходных элементов обучающей выборки равны $1/n$. Следует отметить, что приведенная задача оптимизации является линейной, и двойственная задачи имеет вид:

$$\overline{R}(w) = \min \left\{ c_0 + \frac{1}{n} \sum_{k=1}^{n} c_k \right\}, \quad (13)$$

при ограничениях $c_0,\ c_k \in \mathbb{R},\ k = 1, ..., n,$

$$c_0 + \sum_{k=1}^{n} c_k \mathbf{1}(i \in I_k) \geq l(y_i, \mathbf{x}_i^{(k)}, w), \quad i = 1, ..., N. \tag{14}$$

Здесь c_0, c_k – новые двойственные переменные оптимизации; $\mathbf{1}(D)$ – индикаторная функция, принимающая значение 1, если условие D выполняется. Если предположить, что измерения датчиков различны для каждого обучающего примера, то последние ограничения можно упростить:

$$c_0 + c_k \geq \max_{i \in I_k} l(y_i, \mathbf{x}_i^{(k)}, w). \tag{15}$$

Подставляя эти ограничения в целевую функцию, получим верхнюю границу ожидаемого риска:

$$\overline{R}(w) = \min\left\{\sum_{k=1}^{n} \max_{i \in I_k} l(y_i, \mathbf{x}_i^{(k)}, w)\right\}. \tag{16}$$

Подставляя петлевую функцию потерь в целевую функцию, добавляя регуляризационное слагаемое Тихонова для ограничения множества возможных решений и упрощая задачу, получаем:

$$\overline{R}(w) = \min\left(\frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C \cdot \sum_{k=1}^{n} \xi_k\right), \tag{17}$$

при ограничениях

$$\xi_k \geq 1 - y_k \cdot f(\phi(\mathbf{x}_i^{(k)}), w), \tag{18}$$

$$i \in I_k, \quad \xi_k \geq 0, \quad k = 1, ..., n. \tag{19}$$

Соответствующая двойственная задача оптимизации (лагранжиан) с переменными α_i может быть записана как

$$\max\left(-\frac{1}{2}\sum_{k=1}^{n}\sum_{t=1}^{n}\sum_{i \in I_k}\sum_{j \in I_t} \alpha_i \alpha_j y_k y_t K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}) + \right. \\ \left. + \sum_{k=1}^{n}\sum_{i \in I_k} \alpha_i\right), \tag{20}$$

при ограничениях

$$\sum_{k=1}^{n}\sum_{i \in I_k} \alpha_i y_i = 0, \tag{21}$$

$$0 \leq \sum_{i \in I_k} \alpha_i \leq C, \quad \alpha_i \geq 0, \quad i \in I_k, \quad k = 1, ..., n. \tag{22}$$

Если сравнить полученную задачу со стандартным SVM, то увидим, что переменные α_i ограничены иначе (см. ограничения (22)).

Если предположить, что все точки интервалов, образованные с использованием сетки, различны (т. е. они уникальны для каждого интервала), то целевую функцию (20) можно переписать:

$$\max\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}) + \sum_{i=1}^{N} \alpha_i\right). \tag{23}$$

Ограничение (21) можно переписать аналогично:

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i \in I_k, \tag{24}$$

$$0 \leq \sum_{i \in I_k} \alpha_i \leq C, \quad k = 1, ..., n. \tag{25}$$

Разделяющая функция принимает вид:

$$f(\mathbf{x}) = \sum_{k=1}^{n}\sum_{i \in I_k} \alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}) + b. \tag{26}$$

Полученный SVM не отличается от стандартного SVM с N обучающими элементами за исключением последних ограничений для множителей Лагранжа, которые группируются в соответствии с информацией, получаемой от роботов. В случае одного робота $I_k = \{k\}$, $N = n$, получаем стандартный SVM.

Модификация AdaBoost

Одним из эффективных алгоритмов обучения является AdaBoost [4]. Однако он используется для точных наблюдений, когда обучающая выборка состоит из однозначных примеров. Для улучшения характеристик классификации СГР предлагается модификация алгоритма AdaBoost для случая многозначных наблюдений.

AdaBoost – достаточно общий алгоритм бустинга, который может использоваться в сочетании со многими другими алгоритмами обучения (слабыми классификаторами) для повышения точности их классификации при помощи итерационного процесса. В соответствии с алгоритмом AdaBoost, одинаковые веса $h = (1/n, ..., 1/n)$ изначально назначаются всем примерам. В каждой итерации веса всех неправильно классифицированных примеров увеличиваются, в то время как веса правильно классифицированных примеров уменьшаются (см. Алгоритм). В результате этого слабый классификатор «усиливается» на плохих примерах обучающей выборки. Кроме того, вес φ_t назначается каждому отдельному классификатору. Больший вес назначается более точному классификатору. Распределение весов $h(t)$ модифицируется после каждой итерации. Таким образом, веса стремятся концентрироваться на «плохих» примерах. Композиционный классификатор c в этом алгоритме – весовое голосование T слабых классификаторов. Предполагается ниже, что слабый классификатор c_t – весовая модификация SVM.

Алгоритм AdaBoost

Исходные данные и параметры: Q (число итераций), S (обучающая выборка)

Вычислить: c_t, φ_t, $t = 1, ..., Q$

1: $t \leftarrow 1$; $h_i(1) \leftarrow 1/n$; $i = 1, ..., n$

2: **Цикл по** t

3: Обучить классификатор c_t, используя веса $h(t)$

4: $e(t) \leftarrow \sum_{i:c_t(x_i)\neq y_i} h_i(t)$

5: **Если** $e(t) > 0.5$, **то**

6: $Q \leftarrow t - 1$

7: **Завершение алгоритма**

8: **Иначе**

9: $\varphi_t \leftarrow \frac{1}{2}\ln\left(\frac{1-e(t)}{e(t)}\right)$

10: $h_i(t+1) \leftarrow h_i(t)\cdot\exp\left(-\varphi_t y_i c_t(x_i)\right)$

11: $t \leftarrow t + 1$

12: **Конец цикла, если** $t > Q$

13: $c(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{Q}\varphi_t c_t(\mathbf{x})\right)$

Для модификации алгоритма AdaBoost сначала необходимо записать весовую версию задачи (20)–(22). Предположим, что следующее условие для весов наблюдений выполняется:

$$\sum_{i\in I_k}\pi_i = h_k,\quad k=1,...,n,\quad \sum_{k=1}^{n} h_k = 1. \qquad (27)$$

Двойственная задача для вычисления $\overline{R}(w)$ в этом случае имеет вид:

$$\overline{R}(w) = \min\left(c_0 + \sum_{k=1}^{n} c_k h_k\right), \qquad (28)$$

при ограничениях c_0, $c_k \in \mathbb{R}$, $k = 1, ..., n$ и (14).

Достаточно просто доказать, что прямая задача оптимизации для минимизации верхнего ожидаемого риска имеет вид:

$$\overline{R}(w) = \min\left(\frac{1}{2}\langle\mathbf{w},\mathbf{w}\rangle + C\cdot\sum_{k=1}^{n} h_k\xi_k\right), \qquad (29)$$

при ограничении (19).

Соответствующая двойственная задача оптимизации с переменными α_i отличается от (20)–(22) только ограничениями:

$$0 \le \sum_{i\in I_k}\alpha_i \le h_k C,\quad \alpha_i \ge 0, i\in I_k,\quad k=1,...,n, \qquad (30)$$

где верхняя граница для α_i определяется теперь весом h_k.

Для использования алгоритма AdaBoost определим, как принять решение о классе многозначного наблюдения. Один из путей – применение популярной стратегии, в соответствии с которой множество $\mathbf{A}_k$ принадлежит классу y, если по крайней мере половина его элементов $\mathbf{x}_j^{(k)}$ принадлежат классу y, то есть

$$y_k^* = \arg\max_{y\in\{-1,1\}}\sum_{i\in I_k}\mathbf{1}(c(\mathbf{x}_i^{(k)}) = y). \qquad (31)$$

Важно заметить, что предлагаемый алгоритм бустинга работает с расширенным обучающим множеством, состоящим из N элементов, но веса модифицируются только для многозначных наблюдений, т. е. нет ограничений для весов элементов из каждого многозначного элемента за исключением ограничения (27).

Модификация весов роботов при многозначных наблюдениях

До сих пор рассматривался «наихудший» пессимистический случай, когда предполагалось, что веса полностью неизвестны в рамках одного многозначного измерения, т. е. предполагалось для k-го локального множества весов, обозначаемого $\mathcal{P}_k$, ограничение $\sum_{i\in I_k} \pi_i = h_k$. Теперь предлагается адаптивный алгоритм для сужения локальных множеств весов и использования дополнительной информации о весах роботов, которая определяется ошибками классификации на каждой итерации обучения. Основная идея, лежащая в основе адаптивного алгоритма, заключается в применении интервальной модели Дирихле (ИМД), предложенной в [13] и используемой в представленном алгоритме бустинга для модификации локальных множеств весов на каждом шаге AdaBoost. Эта идея близка идее, используемой в алгоритме, предложенном в [11], где применяется ИМД в AdaBoost для устранения проблемы переобучения.

Предположим, что после построения классификатора на основе n многозначных наблюдений на t-й итерации бустинга имеются $r_i^{(t)}$, корректно классифицированные, и $n - r_i^{(t)}$, ошибочно классифицированные, измерения от i-го робота, $i = 1, ..., T$. Эта информация позволяет нам модифицировать локальные множества. Очевидно предположить, что при наличии большого числа ошибочно классифицированных наблюдений, множества весов $\mathcal{P}_k^{(t)}$ должны расширяться для принятия робастного решения. Более того, оно должно увеличиваться для роботов, измерения которых ошибочно классифицированы.

Рассмотрим кратко ИМД. Пусть $U = \{u_1, ..., u_T\}$ – множество возможных исходов u_j. Предположим что они описываются стандартной мультиномиальной моделью: n наблюдений независимо выбираются из U с вероятностями $\Pr\{u_j\} = p_j$ для $j = 1, ..., T$, где $p_j \geq 0$ и $\sum_{j=1}^{T} p_j = 1$. Тогда ИМД определяется [13] как множество всех распределений Дирихле на вероятностях $p_1, ..., p_T$, параметры которых s и средние значения $\mathbf{q} = (q_1, ..., q_T)$, так что $\mathbf{q}$ принадлежит единичному симплексу размерности T, обозначенному как $S(1, T)$. Параметр s определяет, как быстро верхняя и нижняя границы вероятности события сходятся с накоплением статистических данных. Чем меньше s, тем быстрее сходимость и более рискованное решение, в то время как большие значения s обеспечивают более осторожные решения.

Предлагается рассмотреть правильно классифицированные измерения i-го робота в качестве исходов u_i. Тогда, в соответствии с ИМД, можно записать границы для вероятности $r_i^{(t)}$ правильно классифицированных измерений следующим образом:

$$\frac{r_i^{(t)}}{D^{(t)} + s} \leq p_i \leq \frac{r_i^{(t)} + s}{D^{(t)} + s}, \quad i = 1, ..., T,$$

где $D^{(t)} = \sum_{i=1}^{T} r_i^{(t)}$ – общее число правильно классифицированных измерений на t-й итерации.

Заметим, что перед итерациями бустинга $r_i^{(t)} = D^{(t)} = 0$. Отсюда $0 \leq p_i \leq 1$. Если умножить границы на $h_k(t)$, то получим границы для вероятностей из множества $\mathbf{P}_k^{(t)}$, то есть

$$\frac{r_i^{(t)} h_k(t)}{D^{(t)} + s} \leq \pi_{ki}^{(t)} \leq \frac{\left(r_i^{(t)} + s\right) h_k(t)}{D^{(t)} + s}, \quad i = 1, ..., T.$$

Здесь $\pi_{ki}^{(t)}$ – вес измерения i-го робота на k-м наблюдении t-й итерации, удовлетворяющее условию

$$\sum_{i=1}^{T} \pi_{ki}^{(t)} = h_k(t), \quad k = 1, ..., n.$$

Обозначим для простоты

$$G_i = \frac{\left(r_i^{(t)} + s\right)}{D^{(t)} + s}, \quad F_i = \frac{r_i^{(t)}}{D^{(t)} + s}, \quad i = 1, \ldots, T.$$

Тогда двойственная задача оптимизации для вычисления $\overline{R}(w)$ на t-й итерации имеет вид:

$$\min\left(c_0 + \sum_{k=1}^{n} h_k(t)\left(c_k + \sum_{i=1}^{T}\left(g_{ki}G_i - d_{ki}F_i\right)\right)\right),$$

при ограничениях c_0, $c_k \in \nu$, $g_{kj} \le 0$, $d_{kj} \ge 0$, $j = 1, \ldots, T$, $k = 1, \ldots, n$, и

$$c_0 + \sum_{k=1}^{n} c_k \mathbf{1}(i \in I_k) +$$

$$+ \sum_{k=1}^{n}\sum_{j=1}^{T}(g_{kj} - d_{kj})\mathbf{1}(I_k(j) = i) \ge l(y_i, \mathbf{x}_i^{(k)}, w), i = 1, \ldots, N.$$

Здесь $I_k(j)$ – j-й элемент I_k. Запишем лагранжиан, предполагая, что $l(y_i, \mathbf{x}_i^{(k)}, w)$ – петлевая функция потерь. Он имеет вид:

$$L = \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C \cdot c_0 + C\sum_{k=1}^{n} c_k h_k(t) - \sum_{k=1}^{n}\sum_{i=1}^{T} g_{ki}\lambda_i -$$

$$-\sum_{k=1}^{n}\sum_{i=1}^{T} d_{ki}\mu_i + C\sum_{k=1}^{n} h_k(t)\sum_{i=1}^{T}\left(g_{ki}G_i - d_{ki}F_i\right) -$$

$$-\sum_{k=1}^{n}\sum_{i=1}^{T}\left(\beta_{ki} + \alpha_{ki}\right)(c_0 + c_k + g_{ki} - d_{ki}) -$$

$$-\sum_{k=1}^{n}\sum_{i=1}^{T}\alpha_{ki}(1 - y_k f(\varphi(\mathbf{x}_i^{(k)}), w)).$$

Здесь α_{ki}, μ_i, λ_i, $i = 1, \ldots, T$, $k = 1, \ldots, n$ – множители Лагранжа. Седловая точка может быть найдена, приравнивая производные к нулю. После упрощения получим следующую задачу оптимизации:

$$\max\left(-\frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{i=1}^{T}\sum_{j=1}^{T}\alpha_{ki}\alpha_{lj}y_i y_j K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(l)}) + \sum_{k=1}^{n}\sum_{i=1}^{T}\alpha_{ki}\right),$$

при ограничениях

$$\sum_{k=1}^{n}\sum_{i=1}^{T}\alpha_{ki}y_k = 0,$$

$$\sum_{i=1}^{T}(\beta_{ki} + \alpha_{ki}) = Ch_k(t), \quad k = 1, \ldots, n,$$

$$CF_i \le \sum_{k=1}^{n}(\beta_{ki} + \alpha_{ki}) \le C\sum_{k=1}^{n} G_i.$$

Введем новые переменные $\gamma_{ik} = (\beta_{ki} + \alpha_{ki})/C$. Тогда все ограничения за исключением первого переписываются как

$$F_i \le \sum_{k=1}^{n}\gamma_{ki} \le G_i, \quad \sum_{i=1}^{T}\gamma_{ik} = h_k(t),$$

$$\alpha_{ki} \le C\gamma_{ki}, \quad i = 1, \ldots, T, \quad k = 1, \ldots, n.$$

В итоге получена новая задача оптимизации для вычисления оптимальных значений α_{ki} и γ_{ki}. Интересно увидеть, что ограничения для γ_{ki} повторяют ограничения для π_i. Это важное свойство полученной задачи оптимизации. Оптимальное решение на t-й итерации будет обозначаться как $\alpha_{ki}^{(t)}$.

Разделяющая функция на t-й итерации имеет вид:

$$f^{(t)}(\mathbf{x}) = \sum_{k=1}^{n}\sum_{i=1}^{T}\alpha_{ki}^{(t)} y_k K(\mathbf{x}_i^{(k)}, \mathbf{x}) + b^{(t)}.$$

После подстановки оптимизационной задачи (39)–(44) в AdaBoost (Шаг 3 Алгоритма), получим двойную адаптацию. Первая – модификация весов наблюдений, вторая – изменение множеств весов.

Заключение

В статье предложены три адаптивных минимаксных алгоритма на основе SVM. Первая, главная особенность алгоритмов состоит в том, что они используют множества весов вместо их точных значений, которые обычно применяются во многих алгоритмах классификации. Причиной использования множеств весов является введенное преобразование многозначных обучающих данных, полученных от многих роботов, в обучающие данные с неопределенностью весов в виде этих множеств. Вторая особенность одного из алгоритмов заключается в том, что он основан на использовании ИМД, позволяющей нам уменьшить множества весов, многократно повторяя процедуру классификации. Важным свойством ИМД является то, что она учитывает априорную полную неопределенность весов до получения наблюдений. Третья особенность алгоритмов – их адаптивность. Наборы весов, назначенных роботам, адаптируются к классификаторам. Четвертая особенность заключается в том, что алгоритмы являются робастными, поскольку они используют минимаксную стратегию для работы с эмпирической мерой весового риска при множествах весов.

Следует отметить, что задачи квадратичной оптимизации, которые необходимо решить для построения предложенных классификаторов, аналогичны стандартным задачам оптимизации SVM. Их отличие заключается в дополнительных линейных ограничениях, фактически ограничивающих множества весов. Несмотря на простоту задачи оптимизации, стандартное программное обеспечение, разработанное для SVM во многих пакетах, к сожалению, не может использоваться. Следовательно, для реализации предложенных алгоритмов необходимо разработать соответствующее программное обеспечение.

Работа выполнена при финансовой поддержке РФФИ, грант № 18-29-03250 мк.

СПИСОК ЛИТЕРАТУРЫ

1. **Cowley A., Hsu H.-C., Taylor C.J.** Distributed sensor databases for multi-robot teams // Proc. of the 2004 IEEE Internat. Conf. on Robotics and Automation. 2004. Vol. 1. Pp. 691–696.

2. **Di Caro G.A., Giusti A., Nagi J., Gambardella L.M.** A simple and efficient approach for cooperative incremental learning in robot swarms // Proc. of the 16th Internat. Conf. on Advanced Robotics. 2013. Pp. 1–8.

3. **Du P., Xia J., Zhang W., Tan K., Liu Y., Liu S.** Multiple classifier system for remote sensing image classification: A review // Sensors. 2012. Vol. 120 (4). Pp. 4764–4792.

4. **Freund Y., Schapire R.E.** A decision theoretic generalization of on-line learning and an application to boosting // J. of Computer and System Sciences. 1997. Vol. 550 (1). Pp. 119–139.

5. **Khamis A., Hussein A., Elmogy A.** Multi-robot task allocation: A review of the state-of-the-art // Cooperative Robots and Sensor Networks. Cham: Springer International Publishing, 2015. Vol. 604. Pp. 31–51.

6. **Navarro I., Matia F.** An introduction to swarm robotics // ISRN Robotics. 2013. Article ID 608164. Pp. 1–10.

7. **Pronobis A., Mozos O.M., Caputo B.** SVM-based discriminative accumulation scheme for place recognition // Proc. of the IEEE Internat. Conf. on Robotics and Automation. 2008. Pp. 522–529.

8. **Ravet A., Lacroix S., Hattenberger G., Vandeportaele B.** Learning to combine multi-sensor information for context dependent state estimation // Proc. of the IEEE/RSJ Internat. Conf. on Intelligent Robots and Systems. Tokyo, 2013. Pp. 5221–5226.

9. **Scholkopf B., Smola A.J.** Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, Massachusetts: The MIT Press, 2002.

10. **Tan Y., Zheng Z.-Y.** Research advance in swarm robotics // Defence Technology. 2013. Vol. 9:0. Pp. 18–39.

11. **Utkin L.V.** The imprecise Dirichlet model as a basis for a new boosting classification algorithm // Neurocomputing. 2015. Vol. 1510(3). Pp. 1374–1383.

12. **Utkin L.V.** An imprecise extension of SVM-based machine learning models // Neurocomputing. 2019. Vol. 331. Pp. 18–32.

13. **Walley P.** Inferences from multinomial data: Learning about a bag of marbles // J. of the Royal Statistical Society. Series B. 1996. Vol. 58. Pp. 3–57.

14. **Yuksel S.E., Wilson J.N., Gader P.D.** Twenty 2019s of mixture of experts // IEEE Transactions on Neural Networks and Learning Systems. 2012. Vol. 230 (8). Pp. 1177–1193.

Статья поступила в редакцию 01.02.2019.

REFERENCES

1. **Cowley A., Hsu H.-C., Taylor C.J.** Distributed sensor databases for multi-robot teams. *In Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004, Vol. 1, Pp. 691–696.

2. **Di Caro G.A., Giusti A., Nagi J., Gambardella L.M.** A simple and efficient approach for cooperative incremental learning in robot swarms. *In Proceedings of the 16th*

International Conference on Advanced Robotics, 2013, Pp. 1–8.

3. **Du P., Xia J., Zhang W., Tan K., Liu Y., Liu S.** Multiple classifier system for remote sensing image classification: A review. *Sensors*, 2012, Vol. 120 (4), Pp. 4764–4792.

4. **Freund Y., Schapire R.E.** A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, Vol. 550 (1), Pp. 119–139.

5. **Khamis A., Hussein A., Elmogy A.** Multi-robot task allocation: A review of the state-of-the-art. *In Cooperative Robots and Sensor Networks*. Cham: Springer International Publishing, 2015, Vol. 604, Pp. 31–51.

6. **Navarro I., Matia F.** An introduction to swarm robotics. *ISRN Robotics*, 2013. Article ID 608164, Pp. 1–10.

7. **Pronobis A., Mozos O.M., Caputo B.** SVM-based discriminative accumulation scheme for place recognition. *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, Pp. 522–529.

8. **Ravet A., Lacroix S., Hattenberger G., Vandeportaele B.** Learning to combine multi-sensor information for context dependent state estimation. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, 2013, Pp. 5221–5226.

9. **Scholkopf B., Smola A.J.** *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, Cambridge, Massachusetts, 2002.

10. **Tan Y., Zheng Z.-Y.** Research advance in swarm robotics. *Defence Technology*, 2013, Vol. 9:0, Pp. 18–39.

11. **Utkin L.V.** The imprecise Dirichlet model as a basis for a new boosting classification algorithm. *Neurocomputing*, 2015, Vol. 1510(3), Pp. 1374–1383.

12. **Utkin L.V.** An imprecise extension of SVM-based machine learning models. *Neurocomputing*, 2019, Vol. 331, Pp. 18–32.

13. **Walley P.** Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 1996, Vol. 58, Pp. 3–57.

14. **Yuksel S.E., Wilson J.N., Gader P.D.** Twenty 2019s of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, Vol. 230 (8), Pp. 1177–1193.

Received 01.02.2019.

СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

ПОПОВ Сергей Геннадьевич
POPOV Sergey G.
E-mail: popovserge@spbstu.ru

УТКИН Лев Владимирович
UTKIN Lev V.
E-mail: lev.utkin@gmail.com

ЗАБОРОВСКИЙ Владимир Сергеевич
ZABOROVSKY Vladimir S.
E-mail: vlad2tu@yandex.ru