# Решающие правила для ансамбля из цепей вероятностных классификаторов при решении задач классификации с пересекающимися классами

***А. А. Остапец***
aostapec@mail.ru
МГУ им. М. В. Ломоносова, Россия, г. Москва, Ленинские горы, д. 1

Рассматривается задача классификации с пересекающимися классами. Исследовано применение ансамбля из цепей вероятностных классификаторов с использованием основных типов решающих правил для формирования итоговых предсказаний. Схема решения рассматривается с точки зрения алгебраического подхода. Алгебраический подход заключается в представлении алгоритма решения задачи в виде суперпозиции двух алгоритмов. На первом этапе строится первый алгоритм (распознающий оператор), который в качестве ответа выдает вектор оценок принадлежности к каждому из классов. В качестве распознающих операторов рассматриваются следующие семейства алгоритмов: линейные классификаторы (базовые классификаторы), цепь вероятностных классификаторов из линейных классификаторов и ансамбль из цепей вероятностных классификаторов. На следующем этапе второй алгоритм (решающее правило) трансформирует этот вектор оценок в финальный ответ. Приведен обзор основных типов решающих правил и исследовано их применение для различных распознающих операторов. Экспериментально показана возможность эффективного использования решающих правил, построенных над результатами прогнозов базовых классификаторов.

**Ключевые слова**: *решающие правила; классификация с пересекающимися классами; построение ансамблей; классификация текстов*

**DOI:** 10.21469/22233792.2.3.02

## 1 Введение

Задача классификации с непересекающимися классами является широко распространенной среди задач машинного обучения. В этой задаче каждый объект связан ровно с одним целевым классом. В зависимости от числа непересекающихся классов $\mathcal{L}$ различают задачу бинарной классификации (при $|\mathcal{L}| = 2$) и задачу многоклассовой классификации (при $|\mathcal{L}| > 2$). Задача классификации с пересекающимися классами позволяет объектам относиться к нескольким классам одновременно.

Пусть $\mathcal{X}$ — пространство объектов; $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ — конечное множество классов; $\mathcal{Y} = \{0, 1\}^k$ — множество всех бинарных векторов размерности $k$. Объект $\mathbf{x} \in \mathcal{X}$ описывается вектором признаков $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ и принадлежит некоторому подмножеству классов $L$ из $\mathcal{L}$. Сопоставим каждому подмножеству классов $L$ бинарный вектор $\mathbf{Y} = (y_1, y_2, \ldots, y_k)$, где $y_i = 1 \Leftrightarrow \lambda_i \in L$.

**Определение 1.** *Пусть дан тренировочный набор данных $S = (\mathbf{x}_i, \mathbf{Y}_i), 1 \leqslant i \leqslant n$, состоящий из $n$ объектов ($\mathbf{x}_i \in \mathcal{X}, \mathbf{Y}_i \in \mathcal{Y}$), взятых из неизвестного распределения $D$. Алгоритмом для решения задачи с пересекающимися классами является классификатор $h : \mathcal{X} \to \mathcal{Y}$ , который оптимизирует заданную функцию потерь* [1].

Часто алгоритмы для решения задачи с пересекающимися классами вместо бинарной классификации для каждого класса, определенной выше, представляют собой вещественнозначную функцию $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$. С помощью данной функции для классифицируемого объекта формируется вектор оценок принадлежности $(g_1, \ldots, g_k)$ к классам $\mathcal{L}$. Каждый

такой алгоритм нуждается во втором алгоритме (решающем правиле), который трансформирует этот вектор оценок $(g_1, \ldots, g_k)$ в бинарный вектор $(a_1, \ldots, a_k) \in \mathcal{Y}$. Ненулевые элементы этого вектора — это множество классов, к которым алгоритм относит объект.

## 2 Методы преобразования задачи

Существуют несколько достаточно простых методов преобразования, которые переводят задачу классификации с пересекающимися классами в задачу, к которой могут быть применены существующие алгоритмы многоклассовой классификации. В данной работе, в дальнейшем, рассматривается метод Binary Relevance (BR), а методы Label Powerset (LP) и Error-Correcting Output Code (ECOC) приведены в качестве альтернативных вариантов решения проблемы преобразования задачи.

### 2.1 Label Powerset

Label Powerset — это простой метод, который рассматривает каждое уникальное множество классов в исходной обучающей выборке как один новый класс в преобразованных данных. К преобразованной задаче могут применяться любые алгоритмы многоклассовой классификации. Предсказанный алгоритмом класс в новой задаче однозначно соответствует определенному множеству классов в исходной задаче. Благодаря методу LP также можно осуществлять ранжирование по вероятности исходных классов при предсказании, используя оценки классификатора на новых сформированных классах [2].

Одна из проблем метода LP заключается в том, что после преобразования данных большая часть новых классов содержит очень мало объектов и распределение объектов в новых классах является крайне несбалансированным. Для решения этой проблемы был предложен метод [2]. В этом методе первым делом подбирается порог для отсечения и находятся все классы, которые в преобразованных данных имеют частоту ниже этого порога. Каждый из таких классов заменяется на меньшие по мощности, непересекающиеся подмножества из исходных классов. Каждое из подмножеств должно иметь частоту выше установленного порога в преобразованных данных.

### 2.2 Binary Relevance

Binary Relevance [3] — это один из самых популярных методов преобразования задачи классификации с пересекающимися классами в задачу с непересекающимися классами. Этот метод создает $k$ наборов данных ($k = |\mathcal{L}|$), по одному набору данных на каждый класс. Все новые наборы данных содержат одинаковое число объектов, равное числу объектов в исходной обучающей выборке. В каждом наборе данных $D_{\lambda_j}, 1 \leqslant j \leqslant k$, позитивным классом являются объекты, которые принадлежат классу $\lambda_j$, а негативный класс присваивается всем оставшимся объектам.

На каждом наборе данных обучается бинарный классификатор. На этапе предсказания для объекта берутся предсказания от каждого бинарного классификатора. Итоговым ответом является объединение классов $\lambda_j$, которые бинарные классификаторы определили как позитивные для объекта. Несмотря на то что BR подход используется во многих практических приложениях, он часто критикуется за неявное предположение о независимости исходных классов, которое может не выполняться на реальных данных.

### 2.3 Error-Correcting Output Code

Интересный метод преобразования задачи многоклассовой классификации в несколько задач бинарной классификации был предложен в работе [4]. Этот метод получил название Error-Correcting Output Code. Алгоритм преобразования заключается в кодировании

меток классов бинарными векторами длины $l$. После этого задача нахождения истинного класса для объекта $\mathbf{x} \in \mathcal{X}$ сводится к определению $l$ неизвестных бит кодового слова класса $y(x)$. Для каждого бита $i$ строится бинарный классификатор $f_i$, отделяющий группу классов со значением $+1$ соответствующего бита от классов со значением $-1$. При классификации для объекта $\mathbf{x}$ вычисляется кодовое слово $\mathbf{f}(x) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_l(\mathbf{x}))$ и выбирается ближайший к $\mathbf{f}(x)$ по расстоянию Хэмминга класс. Для задач с пересекающимися классами этот подход можно совместить с подходом LP.

### 2.4 Использование связей между классами

Многие методы преобразования (например, рассмотренный выше BR имеют в себе предположение о независимости классов. Одна из идей для оценки совместного распределения классов была предложена в [5]. Для этого метода необходимо обучить $k$ различных функций ($k = |\mathcal{L}|$) на расширенных признаковых пространствах $\mathcal{X} \times \{0,1\}^{i-1}$, где $y_1, y_2, \ldots, y_{i-1}$ — это оценки принадлежности к классам $\lambda_1, \lambda_2, \ldots, \lambda_{i-1}$;

$$f_i : \mathcal{X} \times \{0,1\}^{i-1} \to [0,1]\,;$$
$$(\mathbf{x}, y_1, y_2, \ldots, y_{i-1}) \to \mathsf{P}(y_i = 1 | \mathbf{x}, y_1, y_2, \ldots, y_{i-1})\,.$$

Здесь $f_i$ могут интерпретироваться как вероятностные классификаторы. Этот подход называется цепью вероятностных классификаторов (Probabilistic Classifier Chain — PCC). В работе [5] описано, как создать ансамбль из таких классификаторов (Ensembled PCC). На каждой итерации новый построенный алгоритм будет отличаться от остальных:

- выбором случайного подмножества объектов для обучения;
- случайной перестановкой порядка классов $\lambda_1, \lambda_2, \ldots, \lambda_k$.

## 3 Решающие правила

Алгебраический подход [6, 7] заключается в представлении алгоритма решения задачи в виде суперпозиции двух алгоритмов:

1) первый алгоритм (распознающий оператор) строит вектор оценок принадлежности классам $(g_1, \ldots, g_k)$, где $g_j$ — оценка принадлежности объекта к $j$-му классу;
2) второй алгоритм (решающее правило) трансформирует вектор оценок $(g_1, \ldots, g_k)$ в бинарный вектор $(a_1, \ldots, a_k) \in \{0,1\}^k$. Ненулевые элементы этого вектора — это классы, к которым алгоритм относит объект.

В работе [8] представлены 4 вида решающих правил. Описание указанных решающих правил приводится ниже.

### 3.1 S-cut

Простейшее решающее правило, возвращающее множество классов, которые получили оценку принадлежности не ниже, чем заданный константный порог $t$:

$$a_i(\mathbf{x}) = \mathbb{I}[g_i(\mathbf{x}) \geqslant t], \forall i \in \mathcal{L}\,.$$

Оптимальное значение порога $t$ можно определить, например, с помощью кросс-валидации.

### 3.2 R-cut

Решающее правило R-cut, в отличие от предыдущего решающего правила, всегда возвращает в качестве ответа множество ровно из $r$ классов с наивысшими оценками:

$$a_i(\mathbf{x}) = \mathbb{I}[\mathrm{rank}(i) \leqslant r], \forall i \in \mathcal{L},$$

где rank$(i)$ — это соответствующая $g_i(\mathbf{x})$ позиция класса $i$ в отсортированным по невозрастанию списке оценок. Как и в случае S-cut, оптимальное значение параметра $r$ можно выбрать с помощью кросс-валидации. Интересный вариант этой стратегии представлен в работе [9], где вместо константного значения $r$ количество классов определяется индивидуально для каждого объекта.

### 3.3 DS-cut

В этом решающем правиле используются несколько параметров $t_i$, каждый из них соответствует позиции $i$, которую класс занимает в отсортированном списке оценок:

$$a_i(\mathbf{x}) = \mathbb{I}[g_i(\mathbf{x}) \geqslant t_{\text{rank}(i)}], \forall i \in \mathcal{L}\,.$$

Здесь необходимо определять не один параметр, как в предыдущих решающих правилах, а $p$ параметров (можно считать, что $t_{(p+1)} = \cdots = t_k = +\infty$). Таким образом, алгоритм может вернуть не более $p$ классов (обычно значение $p$ выбирается равным от 4 до 7).

### 3.4 DSS-cut

Это решающее правило похоже на предыдущее, за исключением того, что абсолютные значения оценок заменены на отношение оценок на конкретных позициях и максимальной оценки в векторе. Эти отношения сравниваются с порогами:

$$a_i(\mathbf{x}) = \mathbb{I}[\frac{g_i(\mathbf{x})}{g_{\max}} \geqslant t_{\text{rank}(i)}], \forall i \in \mathcal{L}\,,$$

где $g_{max}$ — это максимальная оценка в векторе $(g_1, \ldots, g_k)$. В случае если $t_1$ — порог для максимальной оценки — установлен равным 1, то алгоритм всегда возвращает по меньшей мере один класс для каждого объекта. Аналогично предыдущему правилу алгоритм может вернуть не более $p$ классов.

Отметим, что для всех рассмотренных выше правил можно подбирать не глобальные пороги, а пороги для каждого класса в отдельности.

## 4 Эксперименты

### 4.1 Набор данных

Эксперименты проводились на наборе данных, предложенном участникам конкурса Greek Media Monitoring Multilabel Classification (WISE 2014) [10], который проводился на платформе Kaggle летом 2014 г. Данные представляли собой статьи, которые были размещены в греческих средствах массовой информации в период с мая по сентябрь 2013 г. Текст каждой статьи был представлен с использованием модели «мешок слов», после чего было осуществлено TF–IDF (term frequency – inverse document frequency) преобразование. Суть модели «мешок слов» состоит в том, что в ней учитывается только количество вхождений каждого слова в документ, а порядок слов в документе полностью игнорируется. TF-IDF — это статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес слова в этом преобразовании пропорционален количеству употреблений этого слова в документе и обратно пропорционален частоте употребления слова в других документах коллекции.

Таким образом, в исходных данных не предоставлены оригинальные тексты, а только предпросчитанные признаки для каждой статьи. Эксперты в данной задаче вручную

разметили классы для этого набора данных. Каждая статья может принадлежать одному или нескольким классам из 203 доступных.

### 4.2 Функционалы качества

В качестве функционалов качества в данной статье будут рассматриваться 2 варианта:

1) усредненная по всем объектам F-мера;
2) точность классификации.

**Усредненная по всем объектам F-мера (Mean F1-Score, Example-Based F-measure)** часто применяется в задачах с пересекающимися классами. При вычислении этого функционала для каждого объекта вычисляется значение F-меры, а затем все полученные значения усредняются:

$$F_{\text{score}} = \frac{1}{M}\sum_{i=1}^{M} f^i_{\text{score}}, \quad f^i_{\text{score}} = 2\frac{pr}{p+r}$$

где

$$p = \frac{\text{tp}}{\text{tp}+\text{fp}}; \quad r = \frac{\text{tp}}{\text{tp}+\text{fn}};$$

tp — количество истинно-положительных значений для объекта $i$; fp — количество ложно-положительных значений для объекта $i$; fn — количество ложно-отрицательных значений для объекта $i$; $M$ — число объектов в тестовой выборке.

При вычислении данной метрики объект $i$ можно представить множеством классов, к которым этот объект действительно принадлежит. Вторым множеством будет множество классов, которые были предсказаны для этого объекта. Тогда для объекта $i$ характеристики tp, fp и fn вычисляются следующим образом:

1) tp — пересечение двух множеств, описанных выше;
2) fp — предсказанные классы, к которым объект не принадлежит в действительности;
3) fn — классы, к которым объект принадлежит в действительности, но которые отсутствуют в множестве предсказанных классов.

В задачах с пересекающимися классами под **точностью классификации** обычно понимают полное совпадение множеств (subset accuracy): предсказанное множество классов должно полностью совпадать с множеством истинных классов.

Пусть $Y_i^{\text{pred}}$ — бинарный вектор принадлежности $i$-го объекта $k$ классам, полученный алгоритмом, а $Y_i^{\text{true}}$ — бинарный вектор принадлежности $i$-го объекта $k$ классам из тестовой выборки. Тогда данный функционал будет вычисляться следующим образом:

$$\text{Accuracy} = \frac{1}{M}\sum_{i=1}^{M} \text{acc}\left(Y_i^{\text{pred}}, Y_i^{\text{true}}\right),$$

где

$$\text{acc}\left(Y_i^{\text{pred}}, Y_i^{\text{true}}\right) = \begin{cases} 1, & \text{если } Y_i^{\text{pred}} \text{ полностью совпадает с } Y_i^{\text{true}}; \\ 0 & \text{иначе.} \end{cases}$$

### 4.3 Результаты

Для экспериментов были зафиксированы следующие модели:

– логистическая регрессия (ЛР) (использовалась реализация из scikit-learn [11] с параметрами penalty='l1', C=6,0 и tol=0,001;

**Таблица 1** Mean F1-Score для различных решающих правил

| Алгоритм | S-cut | R-cut | DS-cut | DSS-cut |
|---|---|---|---|---|
| Логистическая регрессия | 73,07 | 73,58 | 76,36 | 78,28 |
| Одна модель PCC на основе ЛР | 73,99 | 73,40 | 76,27 | 78,24 |
| Две модели PCC на основе ЛР | 74,52 | 73,68 | 76,68 | 78,32 |
| Три модели PCC на основе ЛР | 74,48 | 73,73 | 76,74 | 78,41 |
| Линейный классификатор (SGD) | 71,80 | 71,53 | 71,12 | 75,52 |
| Одна модель PCC на основе ЛК | 71,96 | 71,46 | 71,06 | 75,41 |
| Две модели PCC на основе ЛК | 72,13 | 71,66 | 71,41 | 75,55 |
| Три модели PCC на основе ЛК | 72,18 | 71,78 | 71,50 | 75,67 |

– линейный классификатор, обученный с помощью метода стохастического градиента (Stochastic Gradient Descent — SGD) (использовалась реализация из scikit-learn [11]: SGDClassifier(loss="modified_huber")).

На основе каждой из этих моделей строились 4 распознающих оператора:

1) исходная модель, обученная с помощью метода BR;
2) цепь вероятностных классификаторов исходной модели;
3) ансамбль из двух цепей вероятностных классификаторов исходной модели;
4) ансамбль из трех цепей вероятностных классификаторов.

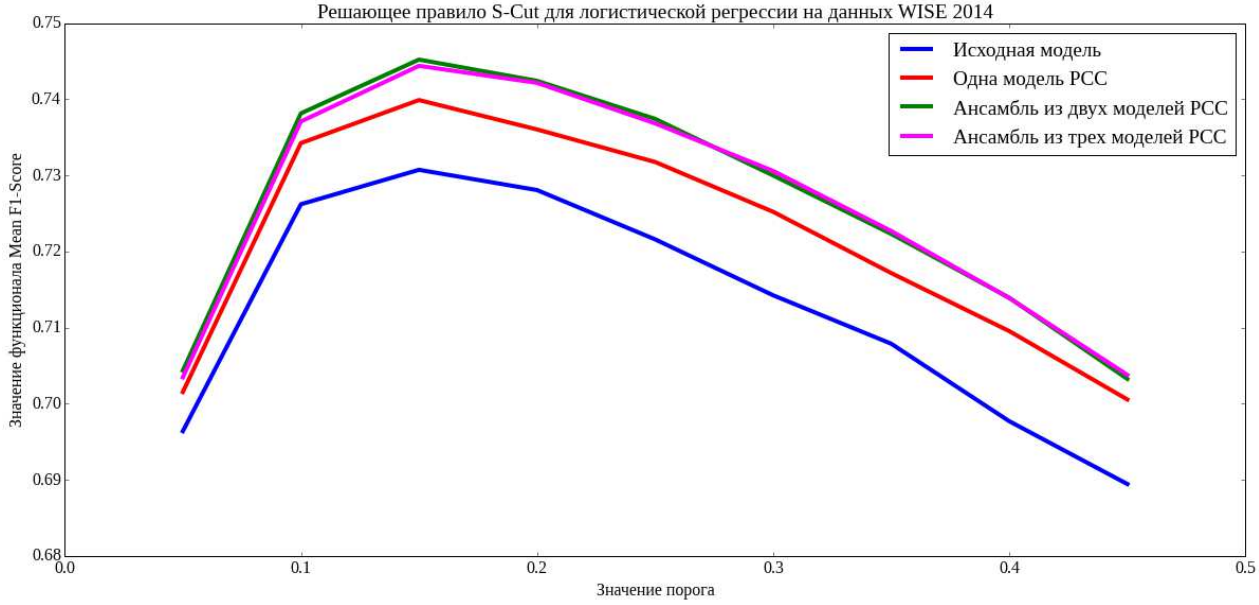
Для всех этих распознающих операторов подбирались пороги для рассмотренных выше решающих правил.

В табл. 1 показаны лучшие значения Mean F1-Score для различных решающих правил (т. е. для каждого решающего правила перебиралось множество различных значений, и лучший результат отображен в таблице). Отметим, что исходную модель для всех решающих правил начинает обходить только ансамбль из двух цепей вероятностных классификаторов.

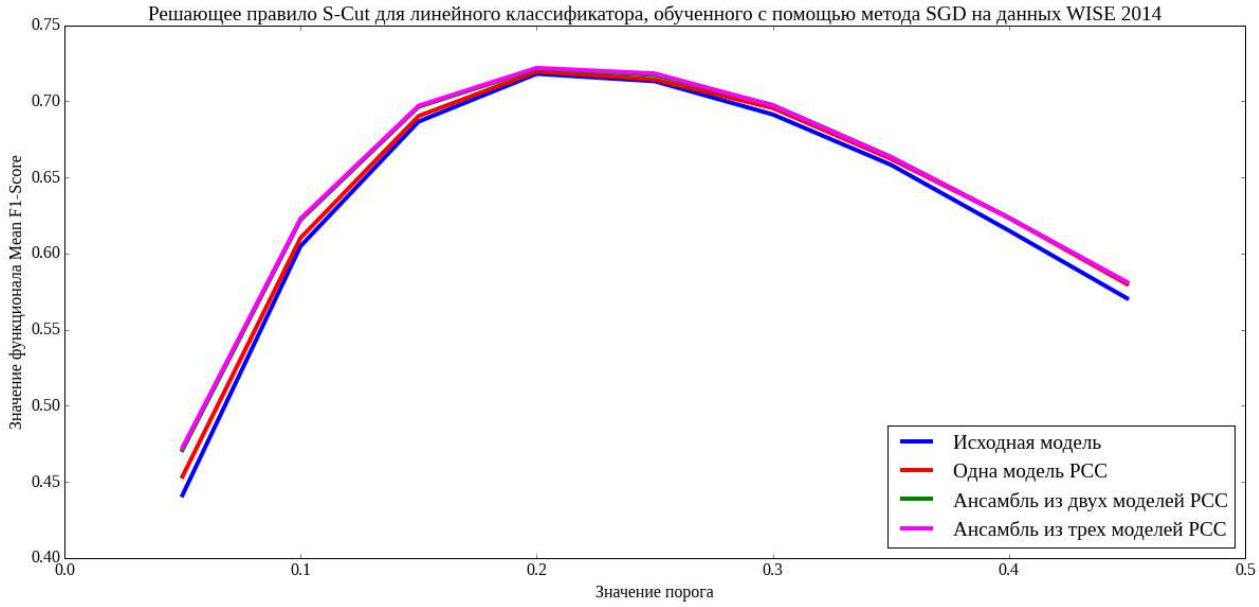
На рис. 1, *а* показана зависимость величины Mean F1-Score решения от порога для решающего правила S-cut для ЛР. Видим, что для этого решающего правила цепь вероятностных классификаторов начинает работать лучше исходной модели даже без использования ансамблей. На рис. 1, *б* показана зависимость качества решения от порога для решающего правила S-cut для линейного классификатора. Здесь качество всех моделей схоже и выигрыш от использования ансамбля значительно меньше.

В табл. 2 показаны лучшие значения точности классификации для различных решающих правил (как и в предыдущем функционале, для каждого решающего правила перебиралось множество различных значений, и лучший результат отображен в таблице).

На рис. 2, *а* показана зависимость величины точности классификации от порога для решающего правила S-cut для ЛР. Как и в предыдущем случае, для этого решающего правила цепь вероятностных классификаторов начинает работать лучше исходной модели даже без использования ансамблей. На рис. 2, *б* показана зависимость величины точности классификации от порога для решающего правила S-cut для линейного классификатора. Аналогично рис. 1, *б* качество всех моделей схоже и выигрыш от использования ансамбля значительно меньше.




(*а*)

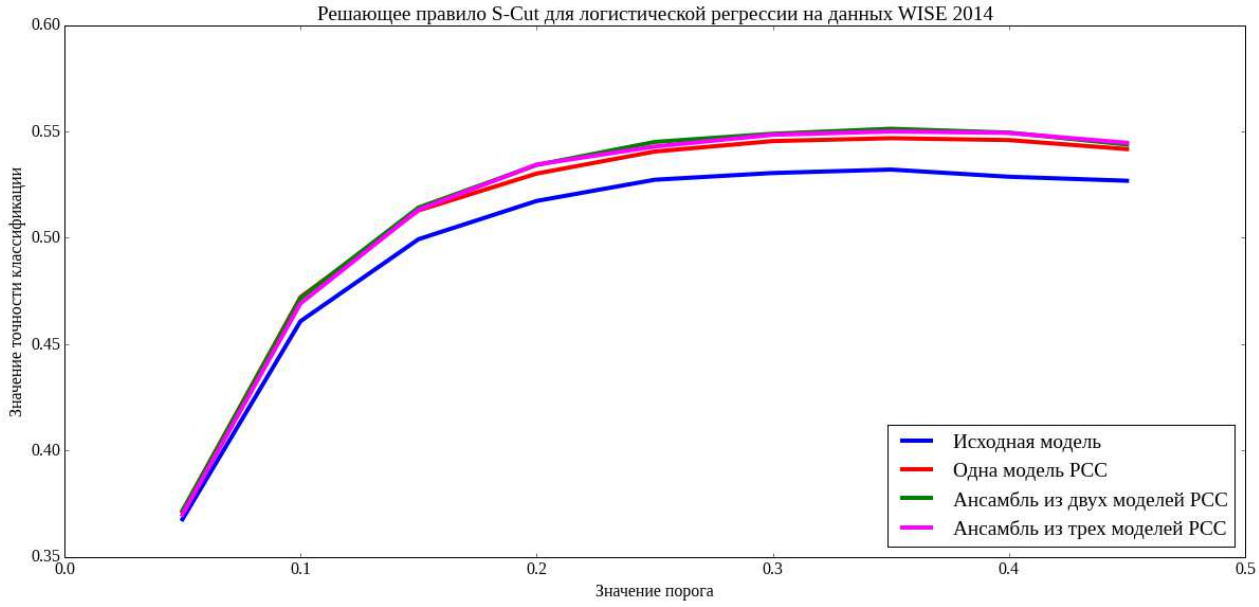



(*б*)

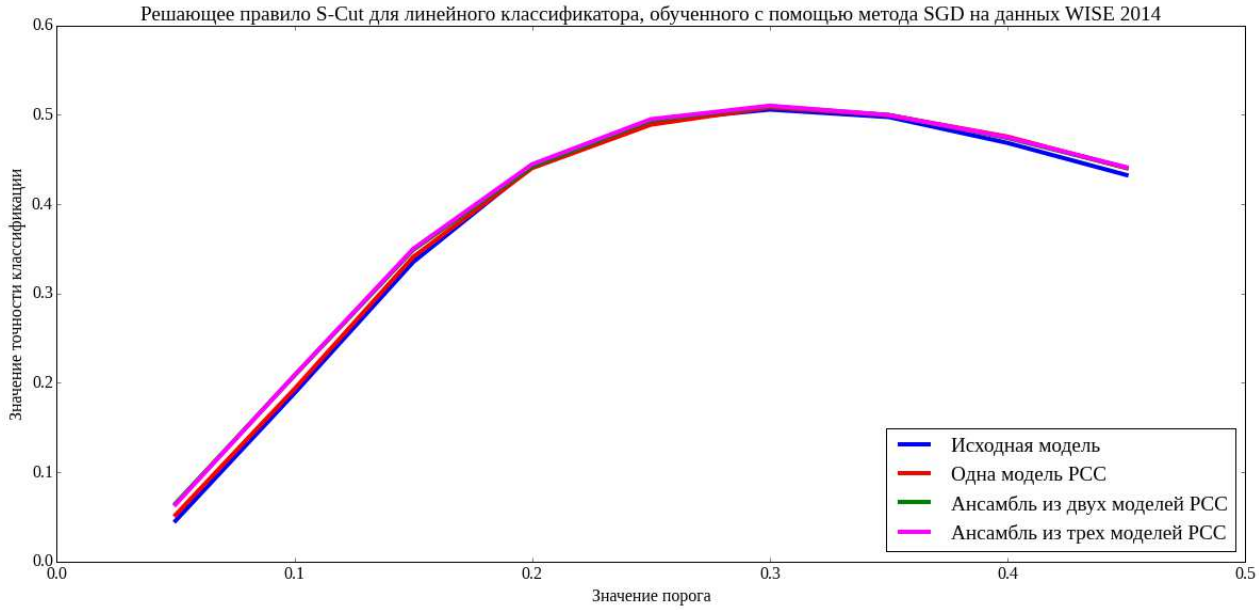
**Рис. 1** Решающее правило S-cut для ЛР (*а*) и линейного классификатора (*б*) (Mean F1-Score)

**Таблица 2** Точность классификации для различных решающих правил

| Алгоритм | S-cut | R-cut | DS-cut | DSS-cut |
|---|---|---|---|---|
| Логистическая регрессия | 52,73 | 58,29 | 53,77 | 59,93 |
| Одна модель PCC на основе ЛР | 54,68 | 58,17 | 54,00 | 59,85 |
| Две модели PCC на основе ЛР | 55,13 | 58,42 | 54,19 | 60,15 |
| Три модели PCC на основе ЛР | 55,20 | 58,50 | 54,25 | 60,21 |
| Линейный классификатор (SGD) | 50,58 | 56,77 | 53,40 | 53,20 |
| Одна модель PCC на основе ЛК | 50,82 | 56,62 | 53,32 | 53,18 |
| Две модели PCC на основе ЛК | 50,94 | 56,89 | 53,51 | 53,55 |
| Три модели PCC на основе ЛК | 51,00 | 56,96 | 53,64 | 53,73 |




($a$)




($б$)

**Рис. 2** Решающее правило S-cut для ЛР ($a$) и линейного классификатора ($б$) (точность классификации)

## 5 Заключение

Показано, что благодаря использованию достаточно простой идеи об учете взаимосвязи между классами и подбора верного решающего правила удается улучшать качество работы базовой модели, обученной с помощью метода BR.

Отдельно отметим, что построение одной цепи вероятностных классификаторов не приводит к улучшению качества работы исходной модели. Необходимый прирост в качестве дает использование ансамбля из двух и более цепей вероятностных классификаторов.

## Литература

[1] *Zhang M. L., Zhou Z. H.* ML-KNN: A lazy learning approach to multi-label learning // Pattern Recogn., 2007. Vol. 40. No. 7. P. 2038–2048.

[2] *Read J.* A pruned problem transformation method for multi-label classification // 2008 New Zealand Computer Science Research Student Conference Proceedings. P. 143–150.

[3] *Godbole S., Sarawagi S.* 2004. Discriminative methods for multi-labeled classification // PAKDD '04: 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining. — Springer, 2008. P. 22–30.

[4] *Dietterich T. G., Bakiri G.* Solving multiclass learning problems via error-correcting output codes // J. Artificial Intell. Res., 1995. Vol. 2. P. 263–286.

[5] *Dembczynski K., Cheng W., Hullermeier E.* Bayes optimal multilabel classification via probabilistic classifier chains. 27th Conference (International) on Machine Learning. — Haifa, Israel, 2010. P. 279–286.

[6] *Журавлёв Ю. И.* Корректные алгебры над множеством некорректных (эвристических) алгоритмов. III // Кибернетика, 1978. №2. С. 35–43.

[7] *Журавлёв Ю. И.* Об алгебраическом подходе к решению задач распознавания // Проблемы кибернетики, 1979. Вып. 33. С. 5–68.

[8] *Wang X. L., Zhao H., Lu B. L.* Enhanced K-nearest neighbour algorithm for large-scale hierarchical multi-label classification // Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification Proceedings, 2011.

[9] *Quevedo J. R., Luaces O., Bahamonde A.* Multilabel classifiers with a probabilistic thresholding strategy // Pattern Recogn., 2012. Vol. 45. No. 2 P. 876–883.

[10] Greek Media Monitoring Multilabel Classification, 2014. `http://www.kaggle.com/c/wise-2014`.

[11] Scikit-learn: Machine learning in Python. `http://scikit-learn.org`.

*Поступила в редакцию 28.08.2016*

# Decision rules for ensembled probabilistic classifier chain for multilabel classification

***A. A. Ostapets***

aostapec@mail.ru

Lomonosov Moscow State University, 1 Leninskie Gory, Moscow, Russia

This work considers using of the main types of decision rules for the multilabel classification task. The algorithm is presented as a superposition of two algorithms: a recognition operator and a decision rule. The recognition operator converts feature vectors of objects to be recognized into scores for each class. This work considers several families of algorithms to be the recognition operator: linear models (base classifiers), probabilistic classifier chain of linear models, and ensembled probabilistic classifier chain. The decision rule converts the scores into the final answers. In this survey, main types of decision rules are described and their performance for several recognition operators is also shown. It is experimentally demonstrated that the quality of the forecast of the proposed composition exceeds the quality of the base classifiers.

**Keywords**: *decision rules; multilabel classification; building ensembles; text classification*

**DOI:** 10.21469/22233792.2.3.02

## References

[1] Zhang, M. L., and Z. H. Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recogn.* 40(7):2038–2048.

[2] Read, J. 2008. A pruned problem transformation method for multi-label classification. *2008 New Zealand Computer Science Research Student Conference Proceedings.* 143–150.

[3] Godbole, S., and S. Sarawagi. 2004. Discriminative methods for multi-labeled classification. *PAKDD '04: 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Springer. 22–30.

[4] Dietterich, T. G., and G. Bakiri. 1995. Solving multiclass learning problems via error-correcting output codes. *J. Artificial Intell. Res.* 2:263–286.

[5] Dembczynski, K., W. Cheng, and E. Hullermeier. 2010. Bayes optimal multilabel classification via probabilistic classifier chains. *27th Conference (International) on Machine Learning.* Haifa, Israel. 279–286.

[6] Zhuravlev, Yu. I. 1978. Korrektnye algebry nad mnozhestvom nekorrektnykh (evristicheskikh) algoritmov. III [Correct algebras for sets of incorrect (heuristic) algorithms. III]. *Kibernetika* [Cybernetics] 2:35–43.

[7] Zhuravlev, Yu. I. 1979. Ob algebraicheskom podkhode k resheniyu zadach raspoznavaniya [An algebraic approach to recognition and classification problems]. *Problemy kibernetiki* [Problems of Cybernetics] 33:5–68.

[8] Wang, X. L., H. Zhao, and B. L. Lu. 2011. Enhanced K-nearest neighbour algorithm for large-scale hierarchical multi-label classification. *Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification Proceedings.*

[9] Quevedo, J. R., O. Luaces, and A. Bahamonde. 2012. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recogn.* 45(2):876–883.

[10] Greek Media Monitoring Multilabel Classification. 2014. Available at: `http://www.kaggle.com/c/wise-2014` (accessed December 13, 2016).

[11] Scikit-learn: Machine learning in Python. Available at: `http://scikit-learn.org` (accessed December 13, 2016).

*Received August 28, 2016*