Гибридный алгоритм классификации текстовых документов на основе анализа внутренней связности текста

И.А. Красников, Н.Н. Никуличев

Введение. В настоящее время во многих прикладных областях, таких как, распознавание речи, распознавание образов, техническая диагностика, биоинформатика, распознавание рукописного ввода, категоризация ввода, хемоинформатика и др., важную роль начинают играть методы машинного обучения и интеллектуального анализа данных. Основное назначение данных методов - анализ, классификация и выявление скрытых закономерностей в больших объемах разнородных сложно структурированных данных [1]. Для решения этих задач разработано множество подходов, имеющих свои преимущества и недостатки, среди которых: метод опорных векторов, метод k-ближайших соседей, нейронные сети, линейная регрессия. Согласно большинству исследований [2-4], одни из лучших результатов показывает наивный байесовский классификатор, основная идея которого заключается в предположении независимости, переданных для классификации признаков, что делает метод довольно простым и точным. В тоже время, неспособность учитывать зависимость результата от сочетания признаков, оказывает существенное влияние на качество классификации в большинстве реальных задач.

Другим подходом, показывающим не менее высокие результаты, является нечеткая классификация [2,8]. При таком методе классификации элементы данных могут принадлежать нескольким классам, связанным с каждым элементом набором степеней принадлежности. Они указывают силу ассоциации между элементом данных и определенным классом. Нечеткая классификация - процесс определения этих степеней принадлежности и использования их, чтобы присвоить элементы данных двум или более классам. В реальных случаях не может быть никаких резких границ между классами, и тогда нечеткая классификация будет лучшим выбором.

Целью настоящего исследования является разработка метода позволяющего объединить два подхода к классификации данных - наивную байесовской модель и нечеткую логику. Основу для классификации данных, составляет байесовская модель, а нечеткая логика используется для перехода от оперирования с независимыми друг от друга элементами данных, к связным данным.

1. Математическая модель гибридного способа классификации текстовых документов

Формально задачу классификации можно описать следующим образом. Предполагается, что алгоритм классификации работает на некотором множестве документов D=$\{d_i\}$. Все множество документов разбивается на непересекающиеся подмножества классов:

$$C=\{Ci\},\bigcup_{d\in Ci}^{\forall i} d=D, Ci \cap Cj=0(i|\neq j)$$

Задачей классификации является определение класса, к которому относится данный документ. Каждому элементу d ставится в соответствие набор признаков $d=\{X_i\}$. Далее применяется алгоритм классификации для выделения документов наиболее соответствующих заданному классу [3].

В основе предложенной системы классификации гипертекстовых документов на естественных языках, лежит наивная байесовская модель, при которой набор переданных для классификации признаков является независимым друг от друга. В общем виде определение наиболее вероятного класса алгоритмом наивной байесовской классификации выглядит следующим образом:

$$C = \arg\ \max_{c\in C} P(C(C_1 o_2 \dots o_n) = \arg\ \max_{c\in C} P(c)\prod P(o_i|c),$$

где C – набор классов, а $o_1 o_2 \dots o_n$ – набор признаков. Классификация сводится к вычислению максимального значения аргумента, при известном наборе независимых признаков $o_1 o_2 \dots o_n$. При этом:

$$P(c)\prod P(o_i|c) = P(C)P(P(o_1|c)P(o_2|c)\dots P(o_n|c).$$

Вычисление вероятности класса *P(C),* при известных признаках $o_1 o_2 \dots o_n$ сводится к следующему:

$$P(C|o_1\ o_2 \dots o_n) = \sum(o_1\ o_2 \dots o_n)+1/\sum(C|A)+\sum A,$$

где *A* – набор известных признаков, полученных при обучении классификатора.

Классификация текста, при этом, выглядит следующим образом:

$$C(T)=\max \sum(t_1 t_2 \dots t_{n1}|C)$$

где *T* – классифицируемый текст, а $t_1, t_2 \dots t_n$ – набор предложений текста. Таким образом, принадлежность текста к тому или иному классу сводится к вычислению максимального значения суммы коэффициентов принадлежности предложений.

Для повышения точности классификации в системе реализован метод предварительного анализа предложений, использующий нечеткую логику. Задачей метода является переход от независимых предложений классифицируемого текста $t_1, t_2 \dots t_n$, к связным. Таким образом, устраняется основной недостаток Байесовской классификации – предположение о независимости признаков используемых при классификации. Нечеткая логика позволяет рассматривать классифицируемый текст в виде связных предложений, т.е. в «контексте».

Пусть имеется множество предложений t_n и множество степеней принадлежности этих предложений n, соответствующих классам C. Преобразование предложения t в степень принадлежности n классу c происходит по формуле (1). Тогда $n_i(c_j)$ степень принадлежности предложения t_i известному классу c_j , $n_{-i}(c_j)$ и $n_{+i}(c_j)$ предыдущее и следующее предложение соответственно. Логика алгоритма представлена на Рис.1 и сводится к набору правил переопределения степеней принадлежности.

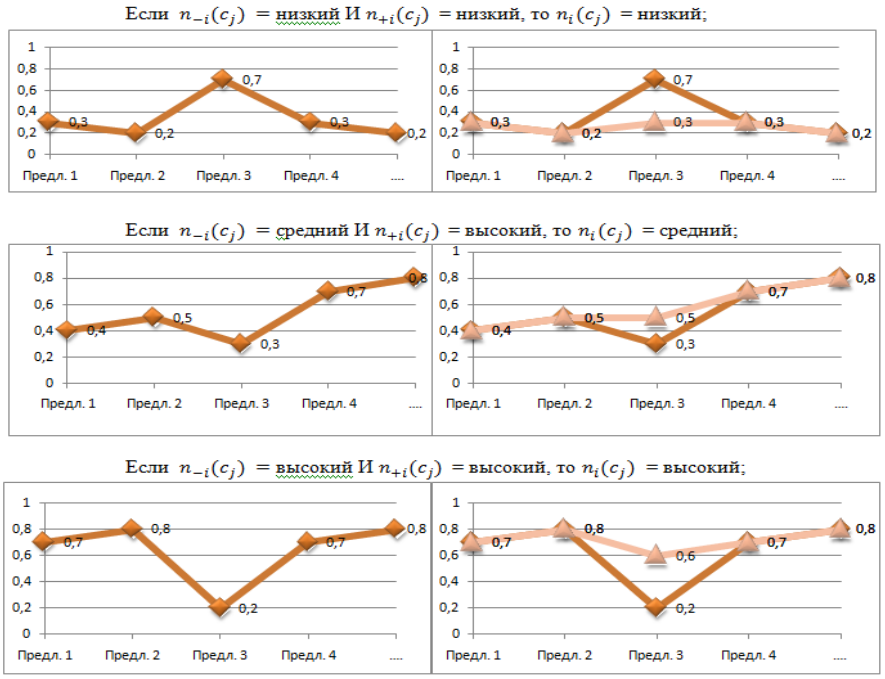

Рисунок 1 - Набор правил переопределения степеней принадлежности

Графики отражают изменения степеней принадлежности, нескольких предложений текста, в зависимости от применения этих правил.

Процесс фаззификации предложений при этом выглядит следующим образом:

$$n(c) = \sum(o_1, o_2 \ldots o_n | c) / \sum(o_1, o_2 \ldots o_n | c_1 + c_2 \ldots c_n), \quad (1)$$

где n – элемент нечеткого множества $N(n_1 n_2 \ldots n_n)$.

2. Алгоритм работы гибридного способа классификации

Логически предложенную гибридную систему классификации документов можно разбить на 3 модуля, это модули обучения системы, модуль байесовской классификации и модуль нечеткого анализа. Алгоритм 1 представляет псевдокод общего алгоритма системы.

Алгоритм 1 Псевдокод общего алгоритма системы

```
1: if (есть (ссылка на страницу)){передать ссылку системе обработки html страниц
2:                     // Система обработки html страниц
3:         function parser($ссылка на страницу) { $html = file_get_html($ссылка на страницу);
4:                 Найти в результирующем массиве $html целевой текст;
5:                 разбить целевой текст  на предложения;
6:                 return $result; }
7:         Передать обучающие выборки функции AddToIndex; Обучить классификатор;
8:                     // Обучение классификатора
```

```
9:      function addToIndex($файл с обучающей выборкой, $класс которому
10:           принадлежит выборка) { создать массив признаков;
11:           загрузить данные из файла в массив признаков;}
12:     foreach($result as $key => $line) {
13:     $результат->classify($line);
14:               // Классификатор
15:               function classify($line){
16:                    $tokens = $this->tokenise($line);
17:                         function tokenise($line) {разбить предложения на слова;
18:                              return массив слов;}
19:          классифицировать  слова $tokens из предложения $line;
20:          преобразовать $line в степень принадлежности $fuzzy; return $result;
21:     $ns->fuzzy($result);
22:          //Нечеткий анализ
23:          function fuzzy($result){
24:               собрать массив степеней принадлежности $fuzzy;
25:               произвести корректировку; return откорректированный массив;}
26:     рассчитать результат;
27:     print "Результат классификации";}
28: else{ print "Ошибка: передана пустая ссылка";}
```

Модуль обучения классификатора представлен в алгоритме 2. Задача модуля сводится к созданию базы признаков для последующей классификации.

Алгоритм 2 Псевдокод обучения классификатора

```
1: function addToIndex($file, $class)
2:      $text = fopen($file, 'r');
3:      while ($line = fgets($text)) {
4:           $tokens= tokenise($line);  // разбить на слова $line, создать массив слов.
5:           foreach($tokens as $token) {
6:                if(в массиве индексов !isset(слова $token принадлежащего классу
7:                  $class)){ добавить запись в массив индексов}}
8:      fclose($text);
```

Алгоритм 3 описывает модуль непосредственной классификации документов. Задачей модуля является классификация переданного текста и создание массива нечетких признаков предложений, для последующего их анализа.

Алгоритм 3 Псевдокод модуля байесовской классификации

```
1: function classify($line)
2:      $tokens = разбить $line на слова, собрать массив слов;
3:      $classScores = array(); // создать массив значений классов.
4:      foreach (classes as $class){
5:           $classScores = добавить все классы из массива классов;
6:           foreach($tokens as $token) {
```

```
7:                    if(в массиве индексов !isset(слова $token принадлежащего классу
8:                      $class)){ счетчик класса $class ++;рассчитать значение $classScores;}}
9:              массив classScores=значение $classScores; }
10:     преобразовать значения счетчика классов в нечеткий вид и записать в массив ns[];
11:     arsort($classScores); return key($classScores) ;
```

В задачи модуля нечеткого анализа входит корректировка предложений в соответствии с логикой, описанной в разделе 1. Схема модуля представлена в алгоритме 4.

Алгоритм 4 Схема Нечеткого анализа

```
1: function fuzzy()
2:      switch($ns[$i])
3:      case 0.6-0.9:
4:              if (предыдущее и следующие значения <=0.5 )
5:              $ns[$i] = 0.4;
6:      case 0.1-0.4:
7:              if(предыдущее и следующие значения >=0.5 )
8:              $ns[$i] = 0.6;
9:      …
10:             записать в массив; break;}
9: return($ns[]) ;
```

Графически предложенный способ классификации представлен на рисунке 2.

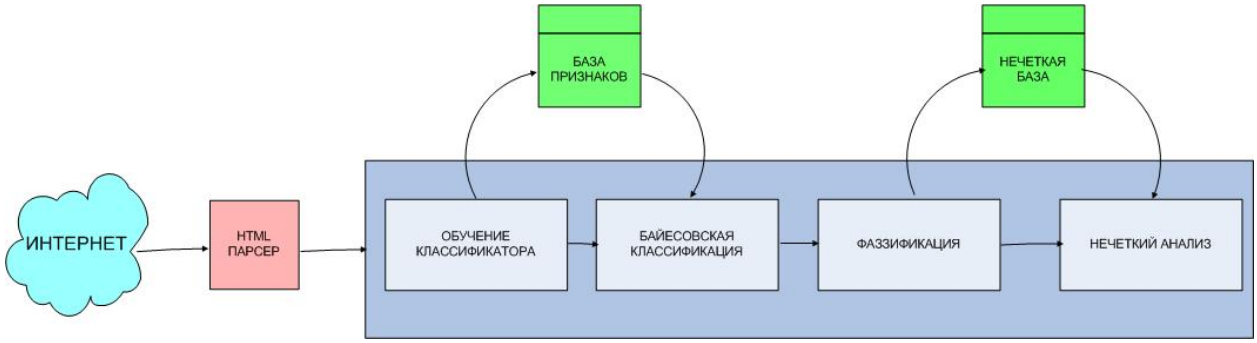

Рис. 2. – Графическое представление гибридного способа классификации

3. Теоретическая оценка вычислительной сложности способа

Для теоретической оценки, использовалась модель вычислений RAM, которая позволяет анализировать алгоритмы машинно-независимым способом [7-9]. Время исполнения алгоритма в RAM-модели вычисляется по общему количеству шагов, требуемых алгоритму для решения некоторого

экземпляра задачи. Постоянные множители, при оценке, были опущены, так как, при возрастании функции они не оказывают существенного влияния.

Вычислительная сложность просмотра каждого элемента в массиве, при обучении классификатора и непосредственной классификации – линейна и равна *n*. Сложность функций двоичного поиска применяемых для просмотра и сравнения элементов в модулях байесовской классификации и нечеткой логики алгоритма, равны *log n*. Вычислительная сложность квадратичной функции сортировки массива используемая при обработке результатов классификации равна n^2. Остальные функции, примененные в алгоритме, в среднем имеют сложность равную константе. Таким образом, функция вычислительной сложности, предложенного способа гибридной классификации текстовых документов, имеет следующий вид:

$$f(n)=n^2+\log n+n+1.$$

Сложность алгоритма квадратична, что является приемлемым, при достаточно больших значениях *n*.

4. Экспериментальное исследование работы гибридного способа классификации

Для проведения эксперимента использовалась коллекция аналитических статей по экономике опубликованных российскими информационными агентствами в период с 05.2012 по 05.2013 гг. В качестве обучающих выборок, использовались коллекции по экономике и политике, общим объемом 10 тыс. слов.

Для сравнительного исследования эффективности предлагаемого способа были проведены эксперименты по классификации разработанным гибридным способом и классическим Байесовским методом. Точность определялась как отношением числа верных определений целевого класса, к общему числу определений [10]. Результаты сравнения способов классификации приведены в таблице №1.

Таблица № 1

Результаты экспериментов

	Наивный байесовский классификатор	Гибридный способ классификации
5 000 слов	87,5%	88,3%
10 000 слов	89%	90,8%
15 000 слов	90,1%	93,7%
20 000 слов	90,7%	94,2%
Общая оценка	89%	92%

Разработанный метод классификации показывает лучшие результаты в сравнении с классическим Байесовским методом на больших (св. 15 тыс. слов) документах, что подтверждают результаты эксперимента.

Заключение. Разработанный гибридный метод классификации текстовых документов позволяет эффективно организовать классификацию коллекций, плохо структурированных данных, больших объемов. Сложность алгоритма является квадратичной, что позволяет показывать высокую производительность при объемах *n,* близких к 1 000 000.

Также предложен подход к устранению основной проблемы Байесовского классификатора - предположению о независимости классифицируемых данных. Использование нечеткой логики позволило выделять «контекст» из классифицируемых данных, и рассматривать текст - связно. Эффективность способа подтверждена результатами экспериментальных исследований.

Предметом исследований автора, проводимых в настоящие время, является разработка метода выявления семантических структур из классифицируемого текста.

Работа выполнена при поддержке РФФИ (проект № 13-07-00951).

Литература

1. Петровский М.И., Глазкова В.В. Алгоритмы машинного обучения для задачи анализа и рубрикации электронных документов. Вычислительные методы и программирование. 2007. Т. 8.№ 2. С. 57-69.

2. Yang Y., Liu X. A re-examination of text categorization methods. // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999 — pp. 42-49.

3. Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. // Proceedings of ECML-98, 10th European Conference on Machine Learning, 1998 — pp. 137-142.

4. Dumais S., Platt J., Heckerman D., Sahami M. Inductive learning algorithms and representations for text categorization. // In Proc. Int. Conf. on Inform. and Knowledge Manage., 1998 — pp. 64-71.

5. M.Seetha, G. Malini Devi, K.V.N.Sunitha An efficient hybrid article swarm optimization for data clustering. [Электронный ресурс] // International Journal of Data Mining & Knowledge Management Process (IJDKP), 2012, Vol.2, No.6, Режим доступа: http://airccse.org/journal/ijdkp/papers/2612ijdkp02.pdf (доступ свободный) – Загл. с экрана. – Яз. англ.

6. Максаков А. В. Исследование способов уменьшения набора характеристик в алгоритмах классификации текстов. Четвертый российский семинар РОМИП. С. 92-100

7. Скиена С. Алгоритмы. Руководство по разработке. – 2 изд. Спб.: БХВ-Петербург, 2011- 720 с.

8. А.П.Рыжов. О качестве классификации объектов на основе нечетких правил. Интеллектуальные системы – 2005. Т.9.В.1-4. С. 253-264.

9. С.П. Алёшин, Е.А. Бородина Нейросетевое распознавание классов в режиме реального времени [Электронный ресурс] // «Инженерный вестник Дона», 2013, №1. – Режим доступа: http://www.ivdon.ru/magazine/archive/n1y2013/1494 (доступ свободный) – Загл. с экрана. – Яз. рус.

10. В.В. Галушка, В.А. Фатхи Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных [Электронный ресурс] // «Инженерный вестник Дона», 2013, №2. – Режим доступа: http://www.ivdon.ru/magazine/archive/n2y2013/1597 (доступ свободный) – Загл. с экрана. – Яз. рус.

11. Гладков Л.А., Гладкова Н.В. Особенности использования нечетких генетических алгоритмов для решения задач оптимизации и управления. //Известия ЮФУ. Технические науки. 2009.№4(93). С.130-136.

12. Курейчик В.В., Сороколетов П.В., Щеглов С.Н. Анализ современного состояния автоматизированных систем приобретения и представления знаний//Известия ЮФУ. Технические науки. 2008. № 9 (86). С. 120-125