Кластеризация текстовых документов из электронной базы публикаций алгоритмом FRiS-Tax*

Н. Г. Загоруйко[1,3], В. Б. Барахнин[2,3], И. А. Борисова[1,3], Д. А. Ткачёв[2]
[1]*Институт математики СО РАН, Новосибирск, Россия*
[2]*Институт вычислительных технологий СО РАН, Новосибирск, Россия*
[3]*Новосибирский государственный университет, Россия*
e-mail: zag@math.nsc.ru, bar@ict.nsc.ru, biamia@mail.ru, relk-tda@yandex.ru

Описывается опыт применения алгоритма FRiS-Tax, основанного на использовании функции конкурентного сходства, в задачах кластеризации текстовых документов. Показано, что для данного класса задач FRiS-алгоритм даёт заметно лучшие результаты по сравнению с классическими алгоритмами кластеризации. Получены апостериорно выбираемые правила для определения весовых коэффициентов при шкалах в формуле вычисления меры сходства на основании предполагаемой достоверности данных. Представлен вариант параллельного выполнения некоторых этапов кластеризации документов с использование FRiS-алгоритма. Приведены количественные оценки времени выполнения процесса, наглядно демонстрирующие преимущества параллельной реализации на разных этапах обработки: при предварительном анализе документов, включающем вычисление мер сходства, а также частично при выполнении непосредственно процесса кластеризации.

Ключевые слова: кластеризация текстовых документов, параллельный алгоритм кластеризации, FRiS-алгоритм.

Введение

В настоящее время, когда накопление самой разнообразной информации происходит гигантскими темпами, важнейшей задачей становится её систематизация и структуризация с целью представления в виде, доступном для понимания и дальнейшего использования. Упоминание об одном из основных подходов к решению этой задачи, дошедшему до нас из глубины веков, встречается еще у Демокрита. В “Письме к учёному соседу” он пишет: “Если тебе, мой друг, нужно разобраться в сложном нагромождении фактов или вещей, ты сначала разложи их на небольшое число куч по похожести. Картина прояснится, и ты поймешь природу этих вещей”. Этот способ познания — создание классификационных структур из множества неструктурированных объектов — широко используется наукой и в наши дни. Его формализация и развитие происходят в рамках научного направления, имеющего целый ряд синонимических названий: таксономия, кластерный анализ, автоматическая классификация, обучение без учителя, самообучение и т. д.

*Работа выполнена при финансовой поддержке РФФИ (гранты № 11-01-00156, 11-07-00561, 12-01-31392, 12-07-00472 и 13-07-00258), Президентской программы “Ведущие научные школы РФ” (грант № НШ 6293.2012.9), Интеграционных проектов СО РАН.

Указанный подход позволяет построить более абстрактную модель информационного массива за счёт сокращения числа рассматриваемых объектов. Вместо изучения каждого отдельного представителя выборки появляется возможность сосредоточить внимание на изучении классов сходных объектов, при этом похожесть в рамках класса позволяет заменять множества объектов из этого класса неким эталонным (идеальным) образцом, реализациями которого эти объекты являются. Если же какой-то класс представляет для исследователя особый интерес, то знание общих для него закономерностей и эталонных представителей даёт возможность расширять этот класс, отыскивая все новые объекты, относящиеся к нему.

Кроме того данный подход даёт возможность ускорять поиск ближайших аналогов в больших базах данных. После проведения таксономии и выделения эталонных представителей каждого класса для нахождения ближайшего аналога нового объекта сначала отыскивается ближайший к нему эталонный представитель, т. е. определяется, к какому из выделенных классов относится новый объект, а затем среди всех объектов данного класса находится ближайший.

Практическим примером задачи, в которой предварительная таксономия исходной выборки позволяет ускорить поиск и улучшить его качество, является задача нахождения по данному документу класса документов, схожих с ним по содержанию. Для того, чтобы решить её, множество документов электронной базы разбиваются на классы по близости в пространстве атрибутов их библиографического описания. При этом для каждого класса выбираются один или несколько эталонных образцов, называемых профилями. Профиль класса представляется некоторым формальным объектом, расположенным в центре класса, или любым представительным объектом, способным характеризовать остальные объекты данного класса. В результате для поиска похожих документов поисковые запросы сначала сравниваются с профилями кластеров, а затем проверяются записи, входящие в кластеры с близкими профилями. Такая задача важна при автоматизации процесса отбора публикаций из электронных баз данных, которые могут быть интересны конкретному исследователю или группе совместно работающих пользователей. Для качественного решения поставленной задачи необходимо, во-первых, определить, как вычислять меру сходства документов, а во-вторых, выбрать алгоритм таксономии, который на основании этой меры будет строить классификации по заданной тематике, пригодные для использования в системе автоматического отбора публикаций.

1. Задание абсолютной меры сходства на множестве документов

В качестве аргументов для определения меры сходства между двумя документами мы используем атрибуты библиографического описания данных документов. Перечислим основные элементы библиографического описания заносимых в картотеку документов:

— авторы;
— заглавие;
— название журнала или издательства;
— год выпуска;
— том, номер, страницы (для публикаций в периодических изданиях);
— аннотация;
— коды классификатора;
— ключевые слова.

Определим меру сходства m двух документов d_1 и d_2 из множества документов D как

$$m : D \times D \to [0, 1],$$

при этом функция m в случае полного сходства принимает значение 1, в случае полного различия — 0. Мера сходства вычисляется по формуле вида

$$m(d_1, d_2) = \sum a_i m_i(d_1, d_2), \tag{1}$$

где i — номер элемента (атрибута) библиографического описания, a_i — весовые коэффициенты, причём $\sum a_i = 1$, $m(d_1, d_2)$ — мера сходства по i-му элементу (иными словами, по i-й шкале).

Поскольку в описываемой ситуации практически все шкалы — номинальные, то мера сходства по i-й шкале определяется следующим образом: если значения i-х атрибутов документов совпадают, то мера сходства равна 1, иначе — 0. При этом необходимо учитывать, что значения атрибутов могут быть составными. В таком случае $m_i = n_{i1}/n_{i0}$, где $n_{i0} = \max\{n_{i0}(d_1), n_{i0}(d_2)\}$, $n_{i0}(d_j)$ — общее количество элементов, составляющих значение i-го атрибута документа d_j, n_{i1} — количество совпадающих элементов. После анализа результатов экспериментальной обработки тестовой выборки алгоритм может быть дополнен апостериорными правилами выбора весовых коэффициентов a_i в формуле (1) на основании предполагаемой апостериорной достоверности данных соответствующей шкалы. Например, полное (или даже почти полное) совпадение значений атрибута “авторы” документов d_1 и d_2 более весомо в случае, когда количество значений этого атрибута в документе d_1 достаточно велико (по сравнению со случаем, когда документ d_1 имеет всего одного автора). В такой ситуации можно увеличивать значение соответствующего весового коэффициента в формуле (1) с одновременным пропорциональным уменьшением других коэффициентов.

2. Понятие конкурентного сходства

Значение предложенной меры сходства зависит только от свойств двух объектов (документов) a и b, между которыми она вычисляется. Однако, если обращаться с вопросом: “Насколько похож объект a на объект b?” к эксперту, то ему обычно требуется дополнительная информация обо всей совокупности рассматриваемых объектов (см., например, [1]). Иначе говоря, вывод о сходстве, например, объекта *кошка* с объектом *корова* различается в случае, когда информационный массив есть множество *лев*, *корова*, и в случае, когда информационный массив есть множество *корова, кобра* (или даже *лев, корова, кобра*). Поэтому один из способов получить более точный и обоснованный ответ на вопрос о сходстве объектов — сформулировать вопрос следующим образом: “Насколько похож объект a на объект b в сравнении с объектом c?” Тем самым мера сходства зависит не только от объектов a и b, но и от контекста — объекта c, выступающего в роли конкурента. В результате формализации идеи о том, что для оценки сходства между объектами необходимо учитывать конкурентную ситуацию, возникает понятие функции конкурентного сходства (FRiS-функции) [2]. В случае заданной абсолютной величины сходства $m(x, y)$ между двумя объектами конкурентное сходство объекта a с объектом b в конкуренции с объектом c вычисляется по следующей формуле:

$$F_{b/c}(a) = \frac{m(a, b) - m(a, c)}{m(a, b) + m(a, c)}.$$

При переходе от сходства между объектами к сходству между объектом и кластером используется тот же принцип. Для оценки конкурентного сходства объекта z с первым кластером учитываются абсолютное сходство $m(z,1)$ z с этим кластером и сходство $m(z,2)$ с конкурирующим вторым кластером. Нормированная величина конкурентного сходства при этом вычисляется по формуле

$$F_{1/2}(z) = \frac{m(z,1) - m(z,2)}{m(z,1) + m(z,2)}.$$

В качестве величины сходства объекта z с кластером могут использоваться величина сходства объекта z с ближайшим к нему объектом из этого кластера либо величина сходства данного объекта с типичным представителем (эталоном) данного кластера.

Значения FRiS-функции меняются в пределах от -1 до $+1$. Если объект z совпадает с эталоном первого кластера, то $F_{1/2}(z) = 1$. При $m(z,1) = m(z,2)$ объект z одинаково похож (или не похож) на оба кластера, тогда значение $F_{1/2}(z) = 0$. При совпадении объекта z с эталоном второго кластера его несходство с первым кластером максимально и равно $F_{1/2}(z) = -1$. Определённая таким способом функция конкурентного сходства хорошо согласуется с человеческими механизмами восприятия сходства и различия. Поэтому предполагается, что, будучи применённой для решения задачи группировки электронных документов, она позволит получить решения лучшего качества, чем при использовании стандартных мер похожести. Остановимся подробнее на описании алгоритма FRiS-Tax [3], использующего FRiS-функцию в процессе построения таксономии.

3. Алгоритм FRiS-Tax

Целью работы данного алгоритма, как и большинства алгоритмов таксономии, является разбиение всего множества объектов выборки A на линейно разделимые кластеры похожих между собой объектов, которые затем объединяются в классы более сложных форм. Причём под похожестью в данном случае понимается конкурентное сходство с центральным объектом кластера (далее такие объекты будем называть столпами кластеров). Если множество столпов $S = \{s_1, s_2, \ldots, s_k\}$ (где k — число кластеров) уже выбрано, то все объекты выборки распределяются между столпами так, чтобы величина конкурентного сходства объектов со “своими” столпами была максимальной. Нетрудно заметить, что у произвольного объекта $a \in A$ максимальное конкурентное сходство будет с ближайшим к нему столпом s_{a1}. Однако остаётся открытым вопрос о том, какой объект использовать для формирования конкурентной ситуации. В задаче распознавания для этих целей используется ближайший конкурент (объект конкурирующего класса). Однако в задаче таксономии принадлежность объектов к классам заранее не известна. Следующий по близости к a столп s_{a2} не всегда будет из конкурирующего класса, так как каждый класс в общем случае может описываться более чем одним столпом. Поэтому в рассмотрение вводится виртуальный конкурент, сходство с которым для всех объектов выборки равно константе m^*. При этом величина конкурентного сходства объекта a с ближайшим к нему столпом s_{a1} из множества S в сравнении с виртуальным конкурентом записывается следующим образом:

$$F^*_{s_{a1}}(a) = \frac{m(a, s_{a1}) - m^*}{m(a, s_{a1}) + m^*}.$$

Естественно, что в задаче таксономии множество столпов S заранее не задано. Выбираться оно будет таким образом, чтобы средняя величина конкурентного сходства каждого объекта выборки A с ближайшим к нему столпом из множества S была максимальна:

$$\overline{F}(S) = \sum_{a \in A} F^*_{s_{a1}}(a) \to \max_S . \qquad (2)$$

Чем больше эта величина, тем более похожи объекты на свои столпы и тем лучше качество формируемой таксономии. Множество столпов будем наращивать последовательно, выбирая их из числа объектов выборки.

1. Поочередно перебирая все объекты выборки A, выделяем объект a^*, для которого величина $\overline{F}(\{a^*\})$ максимальна, и назначаем его на роль первого столпа s_1.

2. После того как первый столп зафиксирован, на роль второго столпа поочередно назначаются все объекты выборки, не совпадающие с s_1. В качестве второго столпа s_2 выбирается тот объект b^*, который в паре с s_1 обеспечивает максимальное суммарное конкурентное сходство $\overline{F}(\{b^*, s_1\})$.

3. Наращивание числа столпов продолжается по этому же принципу. Если первые i столпов $\{s_1, s_2, \ldots, s_i\}$ уже определены, то на роль s_{i+1}-го столпа выбирается объект $z^* \in A/\{s_1, s_2, \ldots, s_i\}$, обеспечивающий максимум функционалу $\overline{F}(\{z^*, s_1, s_2, \ldots, s_i\})$. Процесс продолжается до тех пор, пока не будет набрано заданное число столпов k.

4. После того как было найдено множество столпов $\{s_1, s_2, \ldots, s_k\}$, вся выборка распределяется между ними. Объект относится к тому кластеру, сходство со столпом которого максимально. Объекты, присоединённые к первому столпу s_1, образуют кластер A_1, объекты, ближайшим для которых оказался столп s_2, образуют кластер A_2 и т. д. В результате получаем разбиение множества объектов A на k кластеров A_1, A_2, $\ldots$, A_k.

5. Поскольку с появлением каждого нового столпа состав кластеров меняется, а положение столпов остаётся фиксированным, то может оказаться так, что для описания кластера A_i наилучшим окажется не столп s_i, а какой-то другой объект из этого кластера. Чтобы улучшить положение эталонных объектов, выполняем следующую процедуру. По очереди перебирая каждый объект a_1 кластера A_1 и вычисляя для него значение $\overline{F}(\{a_1, s_2, \ldots, s_k\})$ по формуле (2), находим такое новое положение столпа s^*_1, которое обеспечивает максимальное конкурентное сходство по всей выборке. По аналогии в кластере A_2 определяется новое положение столпа s^*_2 по максимуму функции $\overline{F}(\{s^*_1, x, \ldots, s_k\})$. Поочередно пересматривая положение столпа s^*_i для каждого следующего кластера A_i, в итоге получаем уточнённое множество столпов $S^* = \{s^*_1, s^*_2, \ldots, s^*_k\}$ и соответствующую уточнённую кластеризацию.

Для формирования классов более сложной формы в алгоритме предусмотрена процедура объединения кластеров в классы. При этом предполагается, что кластеры, относящиеся к разным классам, отделяются друг от друга зонами с пониженной плотностью объектов, а на границе кластеров, относящихся к одному классу, такого понижения плотности нет, объекты выборки там распределены достаточно равномерно. Общая схема оценки плотности расположения объектов на границах кластеров выглядит следующим образом.

1. Каждую пару кластеров A_i и A_j проверяют на наличие объектов, которые находятся около разделяющей их границы (в зоне конкуренции). Объект a считается относящимся к зоне конкуренции кластеров A_i и A_j, если выполняются условия:

а) столпы кластеров A_i и A_j являются двумя ближайшими к нему столпами;

б) абсолютная величина FRiS-функции для данного объекта меньше некоторого порога F^*. Кандидатами на объединение считаются те пары кластеров, зоны конкуренции которых не пусты.

2. За расстояние D_{ij} между кластерами A_i и A_j принимается минимальное расстояние между двумя объектами a и b, попавшими в зону конкуренции и принадлежащими разным кластерам.

3. Для этих объектов a из A_i и b из A_j, определяются расстояния D_a и D_b от каждого из них до ближайшего соседа.

4. Кластеры A_i и A_j считаются принадлежащими одному классу, если значения трёх величин D_{ij}, D_a и D_b мало отличаются друг от друга. Например, может проверяться следующее условие:

$$(D_a < \alpha D_b) \wedge (D_b < \alpha D_a) \wedge (D_{ij} < \alpha(D_a + D_b)/2), \quad \alpha > 1.$$

4. Таксономия текстовых документов

В качестве практического примера использования описанного алгоритма рассмотрена задача автоматизации процесса отбора публикаций из электронных баз данных, которые могут представлять интерес для конкретного исследователя или группы совместно работающих исследователей. Решение этой задачи было разбито на ряд этапов.

4.1. Оценка соответствия алгоритма FRiS-Tax поставленной задаче

Тестирование проводилась на электронной базе данных "Сибирского математического журнала", содержащей библиографические описания статей журнала, опубликованных в период с 2000 по 2005 годы. Статьям в указанной базе данных, кроме стандартных атрибутов (название, автор, год издания и т. п.), приписаны соответствующие коды классификатора из "Классификации математических сущностей" (MSC-2000). На первом этапе тестирования рассматривалось пространство документов с одним атрибутом — кодом классификатора MSC-2000 (обычно документу приписано три или более кодов). Поскольку совпадение кодов для группы документов является объективным критерием общности тематики данных документов, такую меру можно считать образцовой. Ниже приведены результаты кластеризации базы данных "Сибирского математического журнала" при помощи алгоритма FRiS-Tax и жадного алгоритма [4] (который, как показано в [5], оказался для рассматриваемой задачи эффективнее метода клик и алгоритма Роккио). На гистограммах (рис. 1) представлен состав полученных кластеров. По горизонтальной оси указаны условные номера кластеров (соответствующие тем или иным разделам классификатора MSC-2000, подробнее см. [6]), вертикальная ось — количество документов в кластере. В качестве критерия проверки правильности отнесения публикации к кластеру использовался его код классификатора из MSC-2000. Если коды классификатора столпа кластера содержались в числе кодов классификатора данной записи, то полагалось, что запись была отнесена к кластеру правильно. Как видно, величина шума (отображаемая в верхней части столбиков) при кластеризации FRiS-Tax существенно ниже, чем в случае жадного алгоритма. Более того, разбиение на кластеры более равномерно, а доля одноэлементных кластеров существенно ниже. Недостатком FRiS-алгоритма является несколько бóльшая вычислительная сложность —

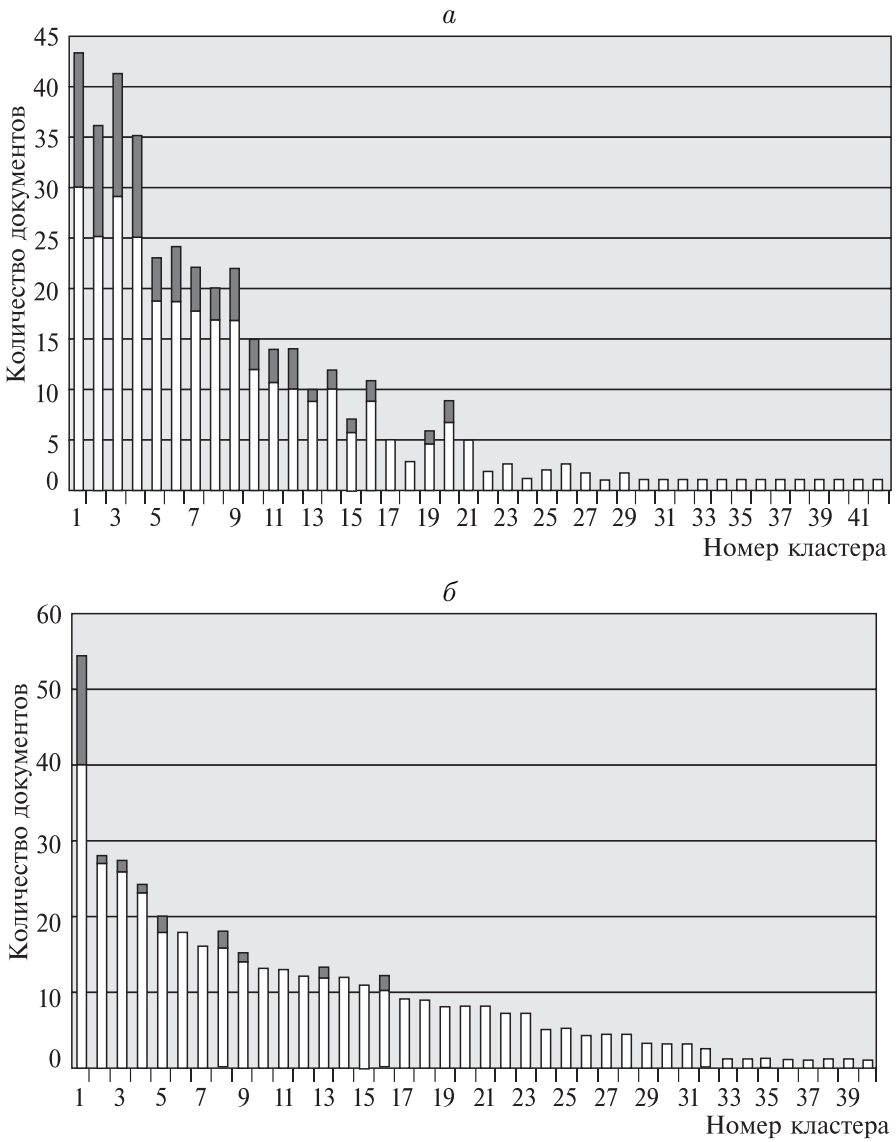

Рис. 1. Качество кластеризации, построенной жадным алгоритмом (*а*) и алгоритмом FRiS-Tax (*б*)

$O(kN^2)$ (k — задаваемое пользователем число кластеров) по сравнению с $O(N^2)$ в случае жадного алгоритма. Однако при кластеризации крупных баз данных такое увеличение сложности становится не столь существенным, к тому же для создания системы, автоматизирующей процесс отбора научных публикаций, кластеризацию базы данных требуется проводить только один раз.

4.2. Экспериментальная настройка коэффициентов в формуле меры сходства

Для задания меры сходства на множестве документов применялась формула (1), где в качестве шкал использовались следующие атрибуты библиографического описания:

— авторы;
— ключевые слова;
— аннотация.

Поскольку сравнение аннотаций в явном виде (т. е. текстовых строк), очевидно, бессмысленно, то как отдельная подзадача решался вопрос выделения терминов из общего текста аннотаций. В качестве источника терминов использован тезаурус [7], построенный на основе "Математической энциклопедии". Тем самым аннотации можно сравнивать между собой как прочие составные атрибуты. Кроме того, при задании меры принят во внимание тот факт, что значения весовых коэффициентов в формуле (1) определяются предполагаемой апостериорной достоверностью данных соответствующей шкалы, и в определённых случаях один из коэффициентов может быть увеличен с пропорциональным уменьшением остальных. Для определения весового коэффициента при каждом из атрибутов проведена кластеризация выборок из базы данных "Сибирского математического журнала". Были рассмотрены выборки различной мощности, а в качестве критерия правильности применялся результат кластеризации, полученный с мерой, основанной на кодах MSC-2000. Как показал эксперимент (подробнее см. [6]), наибольшее сходство с результатом кластеризации при мере, базирующейся на кодах классификатора, было достигнуто при следующих апостериорных правилах выбора коэффициентов.

1. Если каждый из документов d_1 и d_2 имеет более двух авторов и, как минимум, $2/3$ из числа авторов совпадают, то соответствующий весовой коэффициент при атрибуте "авторы" полагался равным 1.

2. Если каждый из документов d_1 и d_2 содержит более трёх ключевых слов и, как минимум, $3/4$ этих слов совпадают, то соответствующий весовой коэффициент при атрибуте "ключевые слова" полагался равным 1.

3. Если каждый из документов d_1 и d_2 содержит более четырёх терминов тезауруса в аннотации и, как минимум, $3/5$ этих терминов совпадают, то соответствующий весовой коэффициент при атрибуте "аннотация" полагался равным 1.

4. В противных случаях коэффициент при атрибуте "авторы" полагался равным 0.2, а при атрибутах "ключевые слова" и "аннотация" — 0.4.

Следует отметить, что эти правила оказались наиболее подходящими как для жадного, так и для FRiS-алгоритма.

4.3. Распараллеливание FRiS-алгоритма

Хотя добавление нового документа не является задачей с большой вычислительной сложностью при известных метриках, на основе которых выполняется расчёт меры сходства документа с документами из групп известных тематик, данный способ вовлечения документов имеет свой недостаток: нельзя бесконечно долго добавлять новые элементы без учёта параметров всех документов, имеющихся в разделяемой выборке. При повторной кластеризации имеющихся документов удаётся избежать ложного отнесения нового материала к уже сформированной группе, схожей с документом характеристикой, которая, вполне возможно, имеет довольно низкий приоритет. Периодическое выполнение процесса разбиения на кластеры является задачей, сложность которой возрастает с увеличением количества обрабатываемых документов.

В настоящее время большой интерес представляет использование в процессе решения задач кластеризации вычислительных систем, состоящих из множества вычислительных узлов. Но, очевидно, в этом случае необходимо иметь версии алгоритмов, которые могут эффективно работать на системах с параллельной архитектурой.

Алгоритмы, применяемые в настоящее время для задач кластеризации и категоризации документов, являются последовательными и используют имеющиеся вычислительные ресурсы далеко не в полную мощность. В них присутствует ряд ограничений, не позволяющих выполнять процесс обработки данных в несколько вычислительных потоков. Во-первых, строгая последовательность выполняемых операций и обязательное завершение предыдущего логического этапа обработки до начала последующего. Во-вторых, все вычисления производятся на основе данных, полученных при анализе всего массива обрабатываемых документов, что в свою очередь довольно сильно усложняет реализацию, а также повышает накладные расходы, связанные с передачей информации между узлами системы.

Преимуществом FRiS-алгоритма является возможность распараллеливания, показанная в [8].

В целом работа программной системы, разбивающей массив текстовых документов на кластеры, состоит из следующих этапов.

1. *Сбор и загрузка исходных данных.* Это действие зависит от источника данных, которыми могут быть данные в текстовых форматах, хорошо поддающиеся обработке и загрузке, импорт из внешних веб-ресурсов, который зависит от производительности внешнего веб-сервера и пропускной способности сети, либо импорт из форматов, обработка которых имеет некоторую специфику, например pdf. Оценивать трудоёмкость данного действия нужно в каждом конкретном случае. В настоящей работе параллельная реализация на этом этапе не использовалась.

2. *Подготовительный этап.* Включает в себя первичную обработку документов. В нашем случае сюда входят процессы выделения ключевых термов и ключевых словосочетаний из текста документов, а также вычисление меры сходства каждого документа с каждым. Отметим, что описанный в предыдущем разделе подход к решению проблемы координатного индексирования заключается в использовании средства анализа на основе тезауруса обрабатываемой предметной области. Однако метод выделения ключевых слов и словосочетаний, основанный на анализе тезауруса предметной области, имеет существенный недостаток: таким способом нельзя производить индексирование корпусов текстов произвольных тематик. Более того, в случаях обработки корпусов текстов достаточно узких тематик требуются весьма подробные тезаурусы, которые можно найти (по крайней мере, в широком доступе) далеко не для всех предметных областей. Подход же, основанный на извлечении ключевых выражений без априорных ограничений, имеет гораздо более универсальный характер, хотя несколько проигрывает в адекватности индексирования. Для решения этой задачи мы использовали результаты морфологического анализа текстов и выделение ключевых словосочетаний по морфологическим шаблонам с использованием программного продукта компании Яндекс (*http://company.yandex.ru/technologies/mystem/*), который является бесплатным для некоммерческих целей. При фильтрации и разборе производился отсев стоп-слов. Ключевые словосочетания отбирались по морфологическим шаблонам с учётом словоформ языка (подробнее см. [9]).

На этом этапе внедрение параллельной обработки может дать существенный выигрыш в производительности, причём действия по анализу содержания являются независимыми друг от друга и не требуют наличия полной информации на всех вычислительных узлах.

3. *Процесс кластеризации.* Исходя из методики работы FRiS-алгоритма, можно сделать вывод, что самым сложным вычислительным процессом является обход всех объ-

ектов выборки и проверка каждого объекта на роль столпа. Понятно, что этот процесс может хорошо выполняться параллельно, хотя и требует наличия информации о расстояниях между объектами на всех вычислительных узлах.

4. *Визуализация результата.* Является вспомогательным действием, облегчающим работу с исходными данными и полученными кластерами. Не требует больших вычислительных мощностей независимо от количества документов в выборке. Оптимизация работы на этом этапе достигается оптимизацией на уровне хранения данных, т. е. на уровне сервера базы данных.

Исходя из вышеизложенного, для этапа подготовки данных и части действий этапа непосредственной кластеризации наиболее приемлемой является параллельная реализация.

5. Экспериментальная проверка

В качестве исходных данных для выполнения анализа и кластеризации использовались материалы трудов международной конференции “Современное состояние наук о Земле”, посвященной памяти В.Е. Хаина, проходившей 1–4 февраля 2011 года в Москве. Исходная выборка включает в себя 488 документов. Выбор исходных данных обусловлен двумя причинами. Во-первых, важно было проверить работу методики выделения ключевых термов из текстов именно геологической тематики без использования тезауруса предметной области, поскольку данная область знаний достаточно сильно насыщена специфическими терминами и определениями, проблемы с анализом которых потенциально могли выявиться в процессе работы. Во-вторых, все документы находятся в рамках одной, уже достаточно узкой, тематики и дальнейшее их разбиение вызывает сложности при использовании мер сходимости, основанных только на библиографических описаниях либо заголовках документов. В первую очередь производилась разбивка исходного множества на установленное количество кластеров — 10, 20, 30. Этот параметр (количество результирующих кластеров) является опциональным и устанавливается перед началом процесса кластеризации. На рис. 2 изображены распределения документов по результирующим кластерам.

Как видно из представленных графиков, несмотря на очевидную близость документов внутри узкой тематики, алгоритм успешно разбил выборки на части. Кластер с номером центроида, равным 0, включает в себя документы, которые не удалось отнести ни к одной из формируемых групп. Количество таких документов в зависимости от количества кластеров варьируется в интервале 7.3–12.7 %, что является приемлемым для работы алгоритмов кластеризации.

Более точно оценить качество разбиения выборки столь узкой тематики можно только с помощью эксперта в предметной области, так как обрабатываемые материалы не имеют кодов принадлежности к уровням того или иного классификатора. В противном случае трудно произвести точную оценку корректности разбиения даже после ознакомления с заголовками и аннотациями статей. Время выполнения определялось следующим образом. Были произведены измерения времени процессов подготовки данных (анализ содержания и вычисление мер близости) и непосредственно кластеризации для 10, 20 и 30 формируемых кластеров на одном вычислительном узле и на нескольких вычислительных узлах для параллельной реализации. причём структура вычислительной сети при выполнении измерений параллельной реализации алгоритма не была однородной и состояла из трёх рабочих станций различной конфигурации (в том числе отлича-

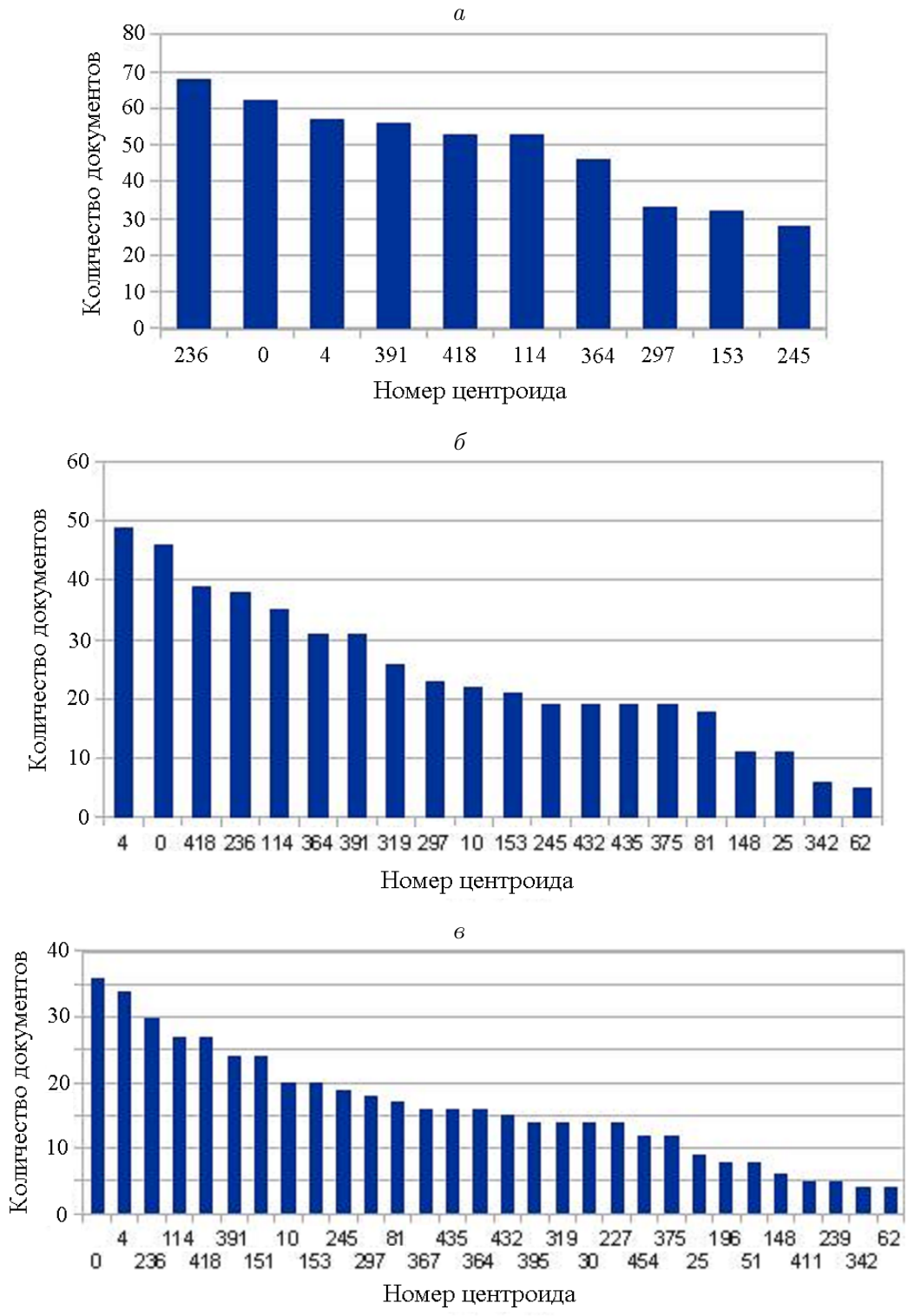

Рис. 2. Количество документов в результирующих кластерах: *а* — 10, *б* — 20, *в* — 30 кластеров

ющихся производительностью), связанных в единую сеть. Данная сеть не может быть бесконечно расширена для ускорения процесса, так как накладные расходы, связанные с передачей информации, будут расти с увеличением количества вычислительных узлов. Результаты измерения времени выполнения последовательной и параллельной реализаций, выполняемых на трёх вычислительных узлах, приведены в таблице.

Время выполнения процесса, с

Этап работы	Количество кластеров		
	10	20	30
Последовательная реализация			
Предварительный анализ данных	290	290	290
Подбор столпов	31	96	200
Итоговое уточнение столпов	8	13	19
Итоговое время	329	399	509
Параллельная реализация			
Предварительный анализ данных	99	99	99
Подбор столпов	11	33	68
Итоговое уточнение столпов	8	13	19
Итоговое время	118	145	186

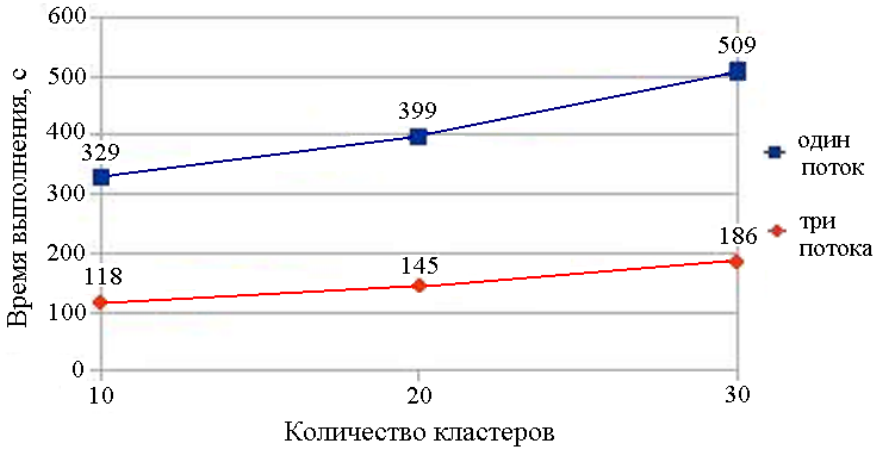

Рис. 3. Время выполнения процесса кластеризации в зависимости от количества вычислительных узлов и кластеров

На рис. 3 представлена зависимость времени выполнения процесса кластеризации от количества вычислительных узлов и кластеров.

Так как разбиение выборки производилась без учёта производительности каждого из вычислительных узлов, сложилась ситуация, что время выполнения не может быть быстрее времени обработки $1/N$ части выборки на самом медленном узле. Тем не менее полученные данные наглядно демонстрируют целесообразность использования параллельной реализации различных стадий обработки данных в процессе кластеризации.

Заключение

В результате проведённого исследования различных алгоритмов кластеризации документов показано, что для задачи разбиения массива записей электронной базы с информацией о научных публикациях на кластеры, содержащие в себе статьи по сходной тематике, лучшие результаты показал алгоритм FRiS-Tax, основанный на использовании функции конкурентного сходства (хотя приемлемые результаты даёт и жадный алгоритм).

Оценка эффективности процесса при использовании распараллеленной версии алгоритма FRiS-Tax по сравнению последовательной реализацией демонстрирует его неоспоримый выигрыш даже несмотря на то, что не все этапы обработки данных в процессе кластеризации могут выполняться параллельно на наборе вычислительных узлов.

Список литературы

[1] Федотов А.М., Барахнин В.Б. Проблемы поиска информации: История и технологии // Вестник НГУ. Информационные технологии. 2009. Т. 7, вып. 2. С. 3–17.

[2] Борисова И.А., Загоруйко Н.Г., Кутненко О.А. Критерии информативности и пригодности подмножества признаков, основанные на функции сходства // Заводская лаборатория. Диагностика материалов. 2008. Т. 74, № 1. С. 68–71.

[3] Борисова И.А. Алгоритм таксономии FRiS-Tax // Научный вестник НГТУ. 2007. № 3. С. 3–12.

[4] Кормен Т., Лейзерсон Ч., Ривест Р.М. Алгоритмы: Построение и анализ М.: МЦНМО, 2001.

[5] Барахнин В.Б., Нехаева В.А., Федотов А.М. О задании меры сходства для кластеризации текстовых документов // Вестник НГУ. Информационные технологии. 2008. Т. 6, вып. 1. С. 3–9.

[6] Шокин Ю.И., Федотов А.М., Барахнин В.Б. Проблемы поиска информации. Новосибирск: Наука, 2010.

[7] Барахнин В.Б., Нехаева В.А. Технология создания тезауруса предметной области на основе предметного указателя энциклопедии // Вычисл. технологии. 2007. Т. 12. Спец. выпуск 2. С. 3–9.

[8] Барахнин В.Б., Ткачёв Д.А. Оценка эффективности метода параллельной реализации процесса кластеризации текстовых документов на основе алгоритма FRiS-Cluster // Вестник НГУ. Информационные технологии. 2012. Т. 10, вып. 4. С. 95–103.

[9] Барахнин В.Б., Ткачёв Д.А. Кластеризация текстовых документов на основе составных ключевых термов // Там же. 2010. Т. 8, вып. 2. С. 5–14.

Поступила в редакцию 19 марта 2013 г.,
с доработки — 29 октября 2013 г.