Компьютерные инструменты в образовании, 2016
№ 1: 24–35
УДК: 51-78, 004.912
http://ipo.spb.ru/journal

КЛАСТЕРИЗАЦИЯ РУССКОЯЗЫЧНЫХ РУКОПИСЕЙ НА ОСНОВЕ ГРАФА ОТНОШЕНИЯ ОСОБЕННОСТЕЙ*

Павлов Владислав Александрович, Дюрдева Полина Сергеевна,
Шалымов Дмитрий Сергеевич

Аннотация

Кластеризация документов — задача объединения текстов по группам таким образом, что все тексты в одной группе обладают некоторыми общими свойствами (принадлежат одному автору, являются текстами одного жанра и др.). Эта задача становится особенно важной по причине стремительно возрастающего количества документов в оцифрованном виде. Для решения задачи кластеризации исследована новая метрика сравнения почерков, основанная на Графах Отношения Особенностей (далее ГОО). Эта метрика успешно зарекомендовала себя при решении текстонезависимой задачи определения автора персидской рукописи на основе почерка. Особенности, основанные на локальных шаблонах, извлекаются из рукописных документов с помощью фильтров Габора и X-Габора (XGabor). Извлеченные особенности формируют ГОО. Исследуется эффективность нескольких наиболее популярных алгоритмов кластеризации для задачи обработки рукописных текстов на русском языке в пространстве ГОО. В работе приведены численные эксперименты, демонстрирующие эффективность предложенной метрики, а также результаты эффективности применения различных алгоритмов кластеризации.

Ключевые слова: *обработка рукописей, русскоязычные тексты, кластеризация текстов, граф отношения особенностей, фильтр Габора.*

1. ВВЕДЕНИЕ

Рукописные тексты представляют собой важный источник информации, поскольку, помимо информативности содержания текста, также содержат особенности, присущие автору, которые могут позволить осуществить графологическую экспертизу [1]. Особенно большой интерес представляют старинные рукописи, являющиеся ценным наследием для историков сегодня. Многие авторы не подписывали свои рукописи, что особенно характерно для древнего времени. Сегодня имеется множество древних рукописей, авторство которых спорно или не установлено вовсе.

Благодаря стремительному развитию технологий значительно возрастает количество документов в оцифрованном виде, в том числе рукописных текстов, которые необходимо эффективно хранить и обрабатывать.

Все это подтверждает актуальность исследования новых алгоритмов для обработки и классификации рукописных текстов. На основе подобных исследований становится воз-

*Работа выполнена при поддержке гранта СПбГУ 6.37.181.2014.

можным создание системы программного обеспечения, способной определять авторство рукописи с определенной точностью.

Последние несколько десятков лет алгоритмы обработки рукописей активно исследуются, и уже получены значительные результаты. Большинство современных систем могут быть классифицированы как онлайн и офлайн системы. Онлайн системы используют информацию, полученную непосредственно в процессе письма. Офлайн системы оперируют с уже готовыми текстами. Системы определения авторства могут быть также поделены на текстозависимые и текстонезависимые. Первые подходят для распознавания авторства множества документов с наперед заданным текстом, в то время как текстонезависимые не зависят от текстов, которые они обрабатывают.

Мы рассматриваем офлайн текстонезависимые системы для русскоязычных рукописных документов. Такие системы могут быть применены к множеству отсканированных рукописей, авторство которых заранее неизвестно. Использование такой системы может позволить определить подмножество документов, написанных похожим почерком (и, вероятно, одним человеком), вычислить число различных стилей письма, определить наиболее вероятного автора рукописи, а также решить более общую задачу — задачу кластеризации рукописей по их почерку.

2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

Русский язык не часто используется для разработки и тестирования систем распознавания авторства. Однако существуют исследования, посвященные этому направлению. В работе [3] исследуется система для автоматического определения языка документа, при этом допускается содержание в документе как рукописного, так и машинописного фрагментов текста. Для извлечения особенностей используется кодовая книга форм (shape codebook). Основная идея данного подхода заключается в анализе специальных криволинейных образцов в тексте документа. Авторам удалось получить очень точные результаты для документов на восьми языках, включая русский.

Исследование распознавания русскоязычного почерка на основе нейронных сетей представил Кулик С.Д. [4]. В качестве входных данных использованы прописные и заглавные буквы русского алфавита. Основной задачей работы является задача классификации рукописей по полу автора. Система показала достойный результат распознавания, составляющий 87 % точности для 650 символов.

Распознавание слитных русских рукописных текстов с использованием аппарата нечеткой логики исследовали Н.С. Исупов и А.В. Кучуганов в работе [5]. Предложенный алгоритм представляет рукописную букву в виде графа. В конце алгоритма для каждого слова формируется нечеткий граф. При распознавании обработанные фрагменты изображения сравниваются с эталонами. Точность распознавания данной системы составила 70%.

Интересны также системы распознавания авторства рукописных документов, настроенные для обработки рукописей на других языках. С нашей точки зрения кажется перспективной система идентификации автора персидского рукописного документа, предложенная в работе [6]. Данная система показала 100-процентную точность распознавания персидских рукописных документов при условии, что представлено достаточное число тренировочных данных. Предлагаемое исследование для русского языка во многом основано на методах, использованных в этой работе. В частности, мы использовали Граф Отношения Особенностей (ГОО) и фильтр Х-Габора, впервые представленные в [6].

3. АГОРИТМ КЛАССИФИКАЦИИ НА ОСНОВЕ ГОО

Для решения задач классификации и кластеризации некоторого множества объектов необходимо определить способ сравнения объектов между собой. Для нашей задачи необходимо определить некоторую меру сходства между двумя отсканированными рукописями (меру сходства между изображениями). Мера, предложенная для персидских текстов [6], может быть использована в качестве таковой. Опишем основные шаги для вычисления данной меры.

3.1. Предобработка

Для эффективной работы алгоритма каждый скан рукописного документа должен быть предобработан. Требуется сегментация на строки. Далее система будет работать с изображениями-строками, а не с целыми сканами документов. Для этого производится бинаризация изображений и применяется необходимая серия морфологических операторов для улучшения линии текста. В нашей системе мы ограничились применением морфологического оператора замыкания.

3.2. Извлечение особенностей

Для каждого входного изображения множество особенностей извлекается с помощью двухмерных фильтров Габора, настроенных на различные направления (ориентации). Такие фильтры позволяют выделять локальные определенно ориентированные паттерны. Фильтр Габора — широко распространенный подход, используемый для решения задач распознавания. Его популярность может быть объяснена тем, что в его работе было найдено определенное сходство с работой зрительной системы человека. Данный фильтр позволяет эффективно производить фильтрацию изображения в частотных и пространственных системах координат. Двумерная функция Габора определяется как произведение синусоидной волны с функцией Гаусса. Функция задается следующим равенством:

$$gabor(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(\frac{-(u^2 + \gamma^2 v^2)}{2\sigma^2}\right) \exp\left(\left(2\pi \frac{u}{\lambda} + \psi\right) i\right), \quad (1)$$

где $u = x\cos\theta + y\sin\theta$ и $v = -x\sin\theta + y\cos\theta$. В соотношении (1) λ представляет собой длину волны синусоидальной функции. В нашем случае эта величина отвечает за толщину линии текста. Параметр θ отвечает за ориентацию нормали к полосам функции Габора. С помощью этой величины задается ориентация локальных паттернов, к которым фильтр наиболее чувствителен. Параметр ψ отвечает за смещение по фазе. В наших экспериментах он приравнен 0. Переменная σ — стандартное отклонение функции Гаусса. Число γ задает эллиптичность функции Габора, и в наших экспериментах мы положили его равным 1. $\frac{\sigma}{\lambda}$ — отношение параметров λ и σ, которое задает диапазон пропускаемых частот. Пример двумерной функции Габора представлен на рис. 1.

Для большей чувствительности к криволинейным паттернам был предложен фильтр Х-Габор (XGabor) [6]. Двумерная функция XGabor может быть определена следующим способом:

$$xgabor(x, y, \lambda, \sigma, r_x, r_y) = \exp\left(\frac{-(x^2 + y^2)}{\sigma^2}\right) \exp\left(\frac{sin\left(\frac{r_x x^2 + r_y y^2}{r_x + r_y}\right)}{\lambda}\right) \quad (2)$$

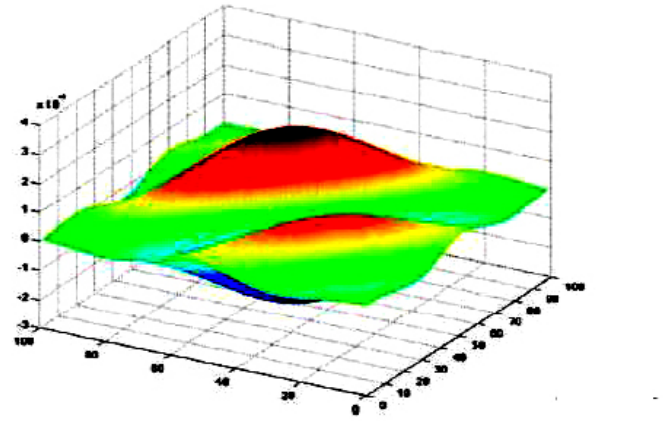

Рис. 1. Двумерная функция Габора

Пояснения требуют параметры r_x и r_y. Данные величины задают темп роста криволинейного паттерна по оси x и y соответственно. Пример двумерной функции XGabor представлен на рис. 2.

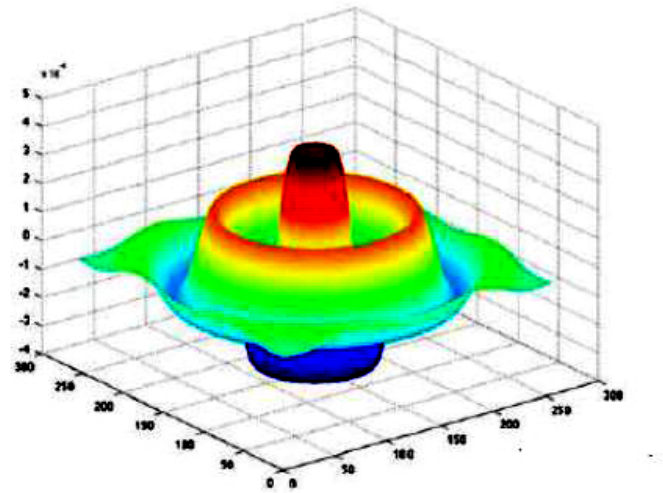

Рис. 2. Двумерная функция XGabor

В нашей системе мы строили вектора особенностей таким же образом, как и в работе [6]. Вектора особенностей размерности l получались после свертки l_1 различно ориентированных фильтров Габора и l_2 XGabor фильтров, настроенных на различные криволинейные паттерны, с входным изображением. В наших экспериментах l_1 принимала значения 45, 36, 8. Величина l_2 приравнивалась к 8 и 36.

3.3. Создание ГОО

Граф отношения особенностей (ГОО) — одно из ключевых понятий предлагаемой системы. Когда все l особенностей получены для всех строк m изображений, вычисляется максимальный разброс M между всеми вычисленными особенностями.

$$M = \max_{s\in\{1..m\},t\in\{1..t\}}(pr(v_s,t)) - \min_{s\in\{1..m\},t\in\{1..t\}}(pr(v_s,t), \tag{3}$$

где $pr(v,i)$ является обозначением для координаты вектора v под номером i.

Для подсчета меры различия между двумя особенностями заполняются пять глобальных переменных M_k, которые подсчитываются на основе M. Пусть $d = \frac{M}{3}$. Тогда M_k подсчитываются согласно 4.

$$M_k = kd - M, \tag{4}$$

где $d = \frac{M}{3}$.

В [6] определены величины r_1, r_2, r_3, r_4, r_5, которые вычисляются для каждой пары особенностей. Каждая r-величина характеризует меру различия пары особенностей. Поведение этих величин в зависимости от аргументов x и y показано на рис. 3.

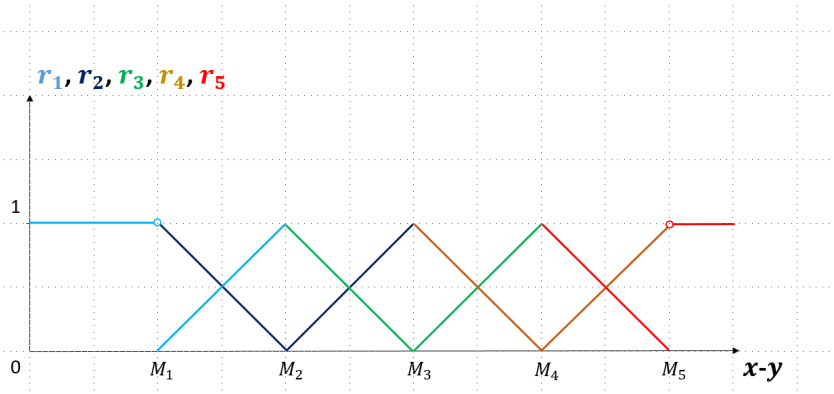

Рис. 3. r-величины

В нашей системе эти величины хранились в матрицах R_1, R_2, R_3, R_4, R_5 для каждого вектора v_s, где $R_k(i,j,s) = r_k(pr(v_s,i), pr(v_s,j))$. После того как R-матрицы вычислены, результирующие матрицы Π_1, Π_2, Π_3, Π_4, Π_5 вычисляются как:

$$\Pi_k(i,j) = \frac{\sum_{s=1}^{l} R_k(i,j,s)}{l}. \tag{5}$$

Теперь может быть построен ГОО. ГОО — ориентированный невзвешенный граф, который содержит не более l вершин, где каждая вершина обозначает некоторую особенность. Множество ребер ГОО G определяется соотношением 6:

$$(x,y) \in E(G) \leftrightarrow \sum_{k=1}^{5}(k-3)\Pi_k(x,y) \geq 1. \tag{6}$$

Наличие ребра от вершины 1 к вершине 2 показывает тенденцию особенности 1 численно превосходить особенность 2 во всех строках входных документов автора. Из определения ГОО следует, что такой граф не может содержать цикл, так как иначе нарушается транзитивность неравенства. Вышеописанный процесс позволяет интерпретировать почерк, задаваемый некоторым множеством изображений, как ГОО.

3.4. Задача классификации

Обозначим ГОО тестового множества документов U и множество графов, полученных на предыдущей стадии алгоритма, как Γ. Для каждого графа G_i из множества Γ подсчитываются меры сходства $S(U, G_i)$. Мера $S(G_1, G_2)$ подсчитывается как число общих путей в графах G_1 и G_2. Специальный алгоритм для эффективного подсчета $S(G_1, G_2)$ предложен в [6]. Его суть заключается в следующем: для всех вершин графов вычисляется длина максимального пути $height$ от вершины до какого-либо листа. На основе полученных величин производится специальная сортировка общих ребер графов по высоте начала ребра. Упорядоченность ребер позволяет вычислить специальную функцию T, необходимую для подсчета меры $S(G_1, G_2)$. Эта функция T может быть вычислена для каждой вершины графа. Если вершина v - лист, то $T(v) = 0$, иначе она получается суммированием всех T величин вершин, инцидентных v. Для подсчета S должна быть вычислена сумма T-величин общих вершин двух графов. Это может быть сделано после обработки ранее отсортированных общих ребер графов в возрастающем порядке.

Стоит заметить, что $S(G_1, G_2)$ является мерой сходства (целая неотрицательная величина), которая возрастает, когда два графа становятся более схожими. В конце этапа решения задачи классификации выбирается граф B, такой что $S(U, B) = \max_{G_i \in \Gamma} S(U, G_i)$. Мы можем построить B для каждого тренировочного множества, тем самым решив задачу классификации.

4. ЧИСЛОВЫЕ ЭКСПЕРИМЕНТЫ

4.1. Классификация рукописей

Алгоритм, кратко изложенный выше, продемонстрировал достойные результаты на персидских рукописных текстах [6]. Мы исследовали эффективность этого алгоритма на русских рукописных текстах. В открытом доступе нами не было найдено баз данных русскоязычных рукописей. Поэтому мы создали свою базу RuHT (Russian Handwritten Texts). Она включает в себя рукописи 30 авторов-носителей языка. Для каждого автора хранятся по 8 строк текста, 3 из которых фиксированы и одинаковы для всех авторов. Примеры рукописей двух различных авторов приведены на рис. 4.

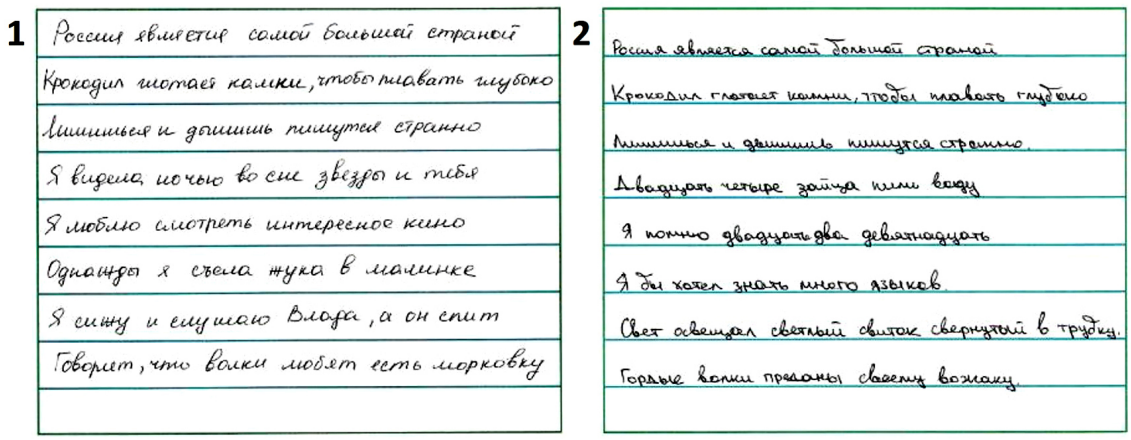

Рис. 4. Примеры рукописей из базы данных RuHT, принадлежащих двум различным авторам

Как было замечено ранее, система требует на вход сегментированные на строки тексты. Поэтому мы произвели сегментацию изображений на строки самостоятельно. Пример полученных после сегментации строк приведен на рис. 5.

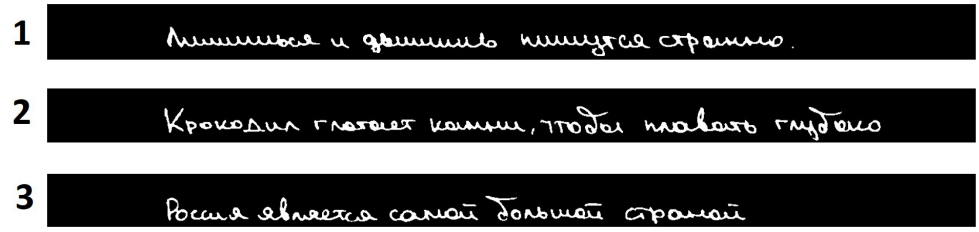

Рис. 5. Примеры строк, полученных после сегментации и предобработки текста

Мы провели ряд экспериментов для тестирования алгоритма на русских текстах. Каждый эксперимент задавался с помощью следующих параметров:

a — число авторов, чьи рукописи участвовали в эксперименте,
p — отношение количества тестовых данных к количеству тренировочных,
l — число извлекаемых особенностей.

Точность распознавания обозначена как c. Эта величина вычисляется как отношение правильно классифицированных документов к общему числу документов, участвовавших в эксперименте, умноженная на 100.

Число строк для каждого автора в каждом эксперименте равно 8.

В *первой серии экспериментов* участвовали рукописи четырех произвольных авторов, поэтому a в данной серии равно 4, p установлено как 3/5 и 5/3, l приравнено к 16 и 32 соответственно.

Во *второй серии экспериментов* участвовали рукописи десяти произвольных авторов ($a = 10$), p установлено как 3/5 и 5/3, l приравнено к 16 и 32.

В *третьей серии экспериментов* $a = 15$, p установлено как 3/5 и 5/3, l приравнено к 32 и 64.

В *четвертой серии экспериментов* $a = 30$, p установлено как 3/5 и 5/3, l приравнено к 64 и 90.

Как видно из таблицы 1, точность алгоритма растет при увеличении объема тренировочных данных и числа извлекаемых особенностей. К сожалению, при росте числа авторов точность алгоритма заметно падает. Однако стоит заметить, что в экспериментах, представленных в [6], использовалось примерно в 5 раз больше данных при решении задачи классификации. Учитывая тот факт, что число данных составляло всего 8 строк для каждого автора в нашей работе, мы считаем, что алгоритм показал себя достойно при классификации русскоязычных документов даже при средней точности, отличающейся от стопроцентной. В то же время при небольшом числе авторов данный алгоритм показал большую точность.

4.2. Кластеризация рукописей

Задача кластеризации известна как задача разбиения входных данных на группы так, чтобы все элементы были как можно более схожи с элементами в своей группе и максимально отличны от элементов из других групп. Обозначим входное множество

Таблица 1. Результаты классификации

№	a	l	p	c
1	5	16	3/5	95
1	5	16	5/3	89
1	5	64	3/5	100
1	5	64	5/3	100
2	10	16	3/5	75
2	10	16	5/3	68
2	10	64	3/5	87
2	10	64	5/3	83
3	15	64	3/5	70
3	15	64	5/3	65
3	15	81	3/5	78
3	15	81	5/3	75
4	28	64	3/5	66
4	28	64	5/3	60
4	28	81	3/5	76
4	28	81	5/3	70

русских рукописных документов как C. В контексте работы наша цель — разбить множество рукописных документов C на множества $C_1, C_2, ..., C_k$ так, чтобы каждый C_i содержал изображения только одного определенного автора. Множества $C_1, C_2, ..., C_k$ называют кластерами, k — параметр, который, как правило, для большинства алгоритмов необходимо оценивать заранее.

Внутреннее качество кластеризации формально можно оценивать целевой функцией $Cost$, значение которой требуется свести к экстремальному (минимум или максимум). Например $Cost(C_1, ..., C_k) \rightarrow \min$. Задачу кластеризации изображений по почерку можно свести к задаче кластеризации графов отношения особенностей по мере их сходства. Стоит заметить, что, ввиду специфичности устройства объектов кластеризации (ГОО), в нашем случае эта функция не будет иметь всех свойств привычной функции расстояния, но будет более похожа на таковую, в отличие от меры S. Обозначим функцию «дистанции» как d. Пусть требуется вычислить «дистанцию» между двумя изображениями S_1 и S_2. Обозначим за L_1 и L_2 строки изображений S_1 и S_2 соответственно. Пусть G_1 и G_2 — графы отношения особенностей, построенные на L_1 и L_2 по алгоритму, представленному в п. 3. Тогда d между S_1 и S_2 определяется как

$$d(S_1, S_2) = \frac{1}{1 + S(G_1, G_2)} \tag{7}$$

Так как $S(G_1, G_2)$ — целочисленная неотрицательная величина, растущая, когда графы становятся все более схожими, d будет равна 1, когда графы полностью различны, и будет стремиться к 0, когда графы становятся более схожими. Чем меньше d, тем более схожи графы и тем они «ближе». В большинстве алгоритмов кластеризации существует понятие центроида — центра масс кластеров. Специфика кластеризуемых элементов также не позволяет использовать традиционные методы подсчета центров кластеров. В нашем случае центроид — это искусственный (не соответствующий ни одному начальному изображению) ГОО, построенный по всем элементам, а точнее, по всем строкам

изображений, входящим в кластер. Так как ГОО выражает собой некоторую статистику, то построенный таким образом «центроид» только улучшает ее представление. Когда определена «дистанция» между двумя документами и определен способ вычисления «центроидов», мы можем применить различные алгоритмы кластеризации.

Мы производили эксперименты со следующими алгоритмами кластеризации: k-means, global k-means [8], Online k-Means, PAM, DBSCAN [9] и агломеративным иерархическим алгоритмом. Для сравнения производительности приведенных выше алгоритмов были рассмотрены следующие внешние метрики качества кластеризации: Purity Measure, Rand Index, Normalized Mutual Information (NMI), F-Measure [11]. Самой простой для вычисления метрикой качества является чистота (purity). Этот показатель эффективен, когда число кластеров невелико. Нормализованная взаимная информация (Normalized Mutual Information) — нормализованный показатель, поэтому его можно использовать для сравнения кластеризации при разном количестве кластеров. Преимуществом коэффициент Рэнда (Rand Index) является учет ложно позитивных решений наравне с ложно негативными. F-мера (F-measure) дополнительно позволяет расставлять разные приоритеты для различного рода ошибjк. Стоит сделать несколько замечаний относительно процедуры тестирования некоторых алгоритмов кластеризации.

4.2.1. Тестирование DBSCAN

Алгоритм DBSCAN сам определяет количество кластеров (параметр k) во время работы, поэтому его указание заранее не требуется. При тестировании алгоритма DBSCAN [9] существуют параметры, которые требуют дополнительного определния. Этот алгоритм параметризуется двумя величинами: ε и $minPoints$. Для каждого параметра были выбраны три стратегии вычисления: равномерная, жадная, усредняющая.

При выборе *равномерной* стратегии определяются максимум и минимум «дистанции» d, вычисленной на всех парах различных графов. Разность между этими величинами делилась на некоторую величину $closeness$, которая задавала требуемую близость элементов при кластеризации. В *жадном* вычислении ε бралось так, чтобы при кластеризации два элемента с наибольшей вероятностью объединялись в один кластер, поэтому в отсортированном по возрастанию массиве всевозможных дистанций Ds выбиралось значение, стоящее на позиции $\frac{2}{3}length(Ds)$. При *усредняющем* вычислении переменной ε присваивалось среднее значение массива Ds. На основе вычисленной величины ε вычислялся параметр $minPoints$. Для каждого элемента вычислялось число соседей в ε окрестности и сохранялось в массиве Ns. Далее $minPoints$ вычислялся по схожим стратегиям, что и ε.

В ходе экспериментов наиболее удачное разбиение входного множества на кластеры было получено при выборе *равномерной* стратегии вычисления каждого параметра.

4.2.2. Тестирование Online K-means

Алгоритм Online K-means работает итеративно, обрабатывая каждый элемент по отдельности и добавляя его в существующий или новый кластер на основе параметра $threshold$. Параметр $threshold$ был вычислен на основе наибольшей дистанции d между элементами с помощью *равномерной* стратегии, описанной в п. 4.2.1. Для алгоритма Online k-means все тексты обрабатывались в произвольном порядке. Процедура кластеризации была произведена несколько раз.

4.2.3. Тестирование остальных алгоритмов

Параметр k был явно задан при работе алгоритмов k-means, global k-means и PAM.

4.2.4. Результаты тестирования

Мы произвели несколько серий экспериментов для каждого алгоритма кластеризации, упомянутого выше. В каждом эксперименте были задействованы рукописи различного числа авторов. Мы обозначили число авторов, чьи рукописи участвовали в процессе кластеризации, через a. В качестве источника рукописных документов была использована база данных RuHT. Восемь строк каждого автора были использованы при кластеризации. В наших экспериментах параметр a был равен 5, 10, 15 и 25 соответственно.

Таблица 2. Результаты кластеризации $*10^2$

a	Algorithm	Purity	RandIndex	NMI	F-Measure
5	k-means	100	100	100	100
5	global k-means	100	100	100	100
5	online k-means	100	78	94	89
5	hierarchy	100	100	100	100
5	DBSCAN	100	89	98	96
5	PAM	100	100	100	100
10	k-means	92	75	86	97
10	global k-means	100	100	100	100
10	online k-means	85	56	92	85
10	hierarchy	80	62	95	90
10	DBSCAN	83	61	89	83
10	PAM	75	58	94	88
15	k-means	64	53	88	90
15	global k-means	89	62	87	91
15	online k-means	61	32	91	88
15	hierarchy	63	26	80	80
15	DBSCAN	73	39	87	82
15	PAM	68	40	93	81
28	k-means	58	25	74	83
28	global k-means	70	38	76	80
28	online k-means	55	20	71	81
28	hierarchy	60	26	80	81
28	DBSCAN	58	24	72	80
28	PAM	65	32	75	85

Таблица 2 показывает результаты применения различных алгоритмов кластеризации на осонове ГОО к входным русскоязычным рукописям. Для пяти кластеров большинство алгоритмов показали 100-процентную точность. Как видно из таблицы, наиболее результативными являются алгоритмы global k-means и PAM. Среди алгоритмов, не требующих входного параметра k, лучшие результаты показал иерархический алгоритм. С возрастанием количества авторов точность кластеризации заметно падает. Как уже было замечено, такая тенденция может быть связана с малым количеством данных для

каждого автора. Однако для объема данных, используемых при экспериментах(8 строк для каждого автора), алгоритмы показали достойные результаты.

5. ЗАКЛЮЧЕНИЕ

В данной работе был использован алгоритм классификации персидских рукописных документов, предложенный B. Helli и M.E. Moghaddam в [6], для решения задачи кластеризации русскоязычных рукописных документов. Была произведена оценка эффективности работы алгоритма на основе ГОО для решения задачи классификации русскоязычных рукописных документов.

ГОО — граф, устройство которого основано на вычислении эмпирических величин, которые вычисляются на основе векторов особенностей, полученных для строк документа. ГОО определяет почерк автора.

Для сравнения двух почерков производится сравнение ГОО с помощью специальной меры схожести. В качестве входных данных были использованы материалы базы данных RuHT, которая была составлена из рукописей 30 авторов, для каждого из которых хранится 8 строк рукописного текста.

Подход на основе ГОО показал высокую точность при решении задачи классификации на 10 авторах. Также мы сравнили результаты работы нескольких популярных алгоритмов кластеризации: k-means, global k-means, Online k-Means, PAM, DBSCAN и агломеративный иерархический алгоритм. Для сравнения результатов кластеризации были использованы различные метрики сравнения: Purity Measure, Rand Index, Normalized Mutual Information, F-Measure.

На небольшом количестве авторов почти все алгоритмы показали высокую точность, на большем количестве авторов (28 авторов) лучшую точность показал алгоритм global k-means.

Список литературы

1. *B. Nevo.* Scientific Aspects Of Graphology: A Handbook. Springfield, IL, 1986.
2. *A. Abbasi, Hsinchun Chen.* Applying authorship analysis to extremistgroup Web forum messages // Intelligent Systems, IEEE, Vol.20(5), 2005.
3. *G. Zhu, X. Yu, Y. Li, D. Doermann.* Language identification for handwritten document images using a shape codebook // Pattern Recognition, Vol. 42, 2009.
4. *Kulik S.D.* Neural Network Model of Artificial Intelligence for Handwriting Recognition // Journal of Theoretical and Applied Information Technology, Vol.73(2), 2015.
5. *Исупов Н.С., Кучуганов А.В.* Распознавание Слитных Рукописных Текстов с Использованием Аппарата Нечеткой Логики // Вестник ИжГТУ, N.1, 2012.
6. *B.Helli, M.E. Moghaddam.* A text-independent Persian writer identification based on feature relation graph (FRG) // Pattern Recognition, Vol.43(6), 2010.
7. *V. Shiv. Naga Prasad and Justin Domke.* Gabor filter visualization. Technical Report, University of Maryland, 2005.
8. *A. Likasa, N. Vlassisb, J. J. Verbeekb.* The global k-means clustering algorithm // Pattern Recognition, Vol. 36(2), 2003.
9. *Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu* A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. AAAI Press, 1996.
10. *A.P. Reynolds, G. Richards and V.J. Rayward-Smith.* The Application of K-medoids and PAM to the Clustering of Rules // Lecture Notes in Computer Science, Vol. 3177, 2004.

11. *C.D. Manning, P. Raghavan, H. Schutze.* Introduction to Information Retrieval. Cambridge University Press, NY, USA, 2008.

RUSSIAN MANUSCRIPTS CLUSTERING BASED ON THE FEATURE RELATION GRAPH (FRG)

Pavlov V. A., Durdeva P. S., Shalymov D. S.

Abstract

Clustering of manuscripts becomes important nowadays because of the rapidly increasing number of documents in digital form. To solve this problem a new metric to compare handwritings based on the Feature Relation Graph (FRG) is investigated. This metric has demonstrated good results for the problem of text-independent writer recognition of Persian manuscripts on the basis of handwriting. Features that are based on local templates are extracted from manuscripts using Gabor and XGabor filters. We study the effectiveness of the most popular clustering algorithms for the problem of Russian manuscripts processing in the phase space of FRG. The paper presents numerical experiments demonstrating the effectiveness of the proposed metrics. The results of the various clustering algorithms are also provided.

Keywords: *Russian manuscripts, clustering, feature relation graph, Gabor filter.*

Павлов Владислав Александрович,
студент кафедры системного программирования математико-механического факультета СПбГУ,
vlad.pavlov24@gmail.com

Дюрдева Полина Сергеевна,
студентка кафедры информационно-аналитических систем математико-механического факультета СПбГУ,
polina.durdeva@yandex.ru

Шалымов Дмитрий Сергеевич,
кандидат физико-математических наук, инженер-исследователь, математико-механический факультет СПбГУ,
dmitry.shalymov@gmail.com