САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра Системного Программирования

Павлов Владислав Александрович

Автоматическое определение авторства рукописных арабографических документов по почерку и их кластеризация на основе Feature Relation Graph

Курсовая работа

Научный руководитель:

к. ф.-м. н. Шалымов Д. С.

Санкт-Петербург

2015

**Оглавление**

1. Введение3
2. Состояние исследуемой области на настоящий момент4
3. Реализация предлагаемой системы6
	1. Алгоритм идентификации6
		1. Предварительная обработка изображений6
		2. Извлечение особенностей изображения7
		3. Конструирование FRG10
		4. Классификация11
	2. Тестирование алгоритма12
		1. Тренировочные данные12
		2. Эксперименты13
	3. Применение алгоритма при кластеризации данных14
4. Результаты17
5. Список литературы18
6. **Введение**

В настоящее время одной из ценностей исследователей в области истории являются дошедшие до наших времен старые рукописи. В древние времена многие писатели не упоминали в своих рукописях о себе. До наших времен дошло множество арабских рукописных текстов, авторство которых либо не установлено, либо сомнительно. Этот факт ставит историков перед проблемой определения авторства арабографического рукописного документа по почерку его создателя. Данная проблема актуальна не только для историков, но и для исследователей в области криминалистики, палеографии и так далее.

Вышеописанная ситуация ставит задачу создания некоторой компьютерной системы для автоматической идентификации автора некоторого арабографического рукописного документа, имея определенный объем информации о её потенциальных создателях. Существование такой системы не решит задачу идентификации автора в полном объеме, но значительно упростит работу специалистов, которые регулярно вынуждены устанавливать авторство арабографических рукописей. Разработка такой системы даст возможность рассмотреть возможные решения другой, более общей задачи – задачи кластеризации документов по авторству. В этом случае входное множество документов должно быть поделено на группы таким образом, чтобы в каждой группе было как можно больше документов одного автора и как можно меньше документов других авторов.

В данной работе был исследован алгоритм идентификации автора арабографического рукописного документа по почерку его создателя на основе графов отношения особенностей[4]. На основе данного алгоритма была разработана компьютерная система для автоматической идентификации автора арабографической рукописи. Алгоритм был протестирован при кластеризации арабографических рукописных документов с использованием различных алгоритмов кластеризации.

1. **Состояние исследуемой области на настоящий момент**

Важно отметить, что арабский язык является особым с точки зрения вариативности стилей письма, способов написания букв, в результате чего для данного языка следует разрабатывать специфичные методы для предварительной обработки и анализа изображений текста. В настоящее время активно ведутся исследования в этой области и достигнуты приличные результаты. Существующие системы для идентификации автора делятся на два типа: online системы и offline системы. Online системы – системы, которые требуют непосредственного присутствия автора. Такие системы используют информацию о давлении пишущего предмета, его скорости движения и подобных характеристиках. В области online систем достигнуты солидные результаты и существуют системы, распознающие авторство с точностью, очень близкой к 100%. Offline системы, наоборот, работают с информацией, полученной с изображения, и только. Также существует разделение на text-dependent и text-independent системы. В первых производится анализ данных заранее известного содержания. Преимуществом таких систем является отсутствие нужды в больших объемах данных. Text-independent системы устанавливают авторство, не зная, что пишется в тексте. Такие системы хороши тем, что не зависят от данных, но, к сожалению, требуют достаточного количества входных данных. Таким образом, в данной работе будут рассмотрены offline text-independent системы.

В [1] была описана система, у которой точность распознавания достигла 96%. Несколько экспериментов было проведено для 40 различных классов авторов. Процесс извлечения векторов особенностей включал в себя извлечение особенностей с использованием фильтра Gabor[6], построение матриц совместного появления. Было произведено сравнение классификатора, основанного на Евклидовой дистанции, с KNN классификатором.

Альтернативная система для арабского языка была предложена в [2]. Метод, использованный в системе, основывался на присутствии определенных шаблонов, ориентации и криволинейности текста в рукописи. Система была протестирована на рукописях различных языков. Точность идентификации автора для арабского языка составила около 92% для 100 авторов.

Система, извлекающая различные типы особенностей, была предложена в [3]. В системе был использован KNN классификатор для Евклидовой дистанции. Для экспериментов была использована база данных, содержащая рукописи более 250 авторов. Эксперименты проводились с использованием всего лишь 2 изображения для каждого писателя. Средняя точность системы составила 95%.

Behzad Helli и Ebrahimi Moghaddam в своей работе[4] описали систему для идентификации автора по почерку для персидского языка. Система использовала особенности, основанные на паттернах текста доля конструирования специальных графов, характеризующих почерк писателя. Особенности были получены при помощи фильтров Gabor и XGabor. Полученные особенности сравнивались между собой с помощью эмпирических методов и строился специальный граф отношения особенностей (feature relation graph, далее FRG). При классификации графы сравнивались между собой с помощью специально разработанных алгоритмов. Предложенная система продемонстрировала почти 100 процентную точность на 80 авторах при отношении объема тренировочных данных к объему тестовых равному 3/2. Такие результаты послужили основной причиной для проектирования будущей системы для арабского языка на основе данной.

1. **Реализация предлагаемой системы**

Разработка системы велась на языке программирования Python.

Далее будут рассмотрены основные этапы реализации системы.

* 1. Алгоритм идентификации

Алгоритм идентификации автора является ключевой составляющей проектируемой системы, поэтому далее он будет описан максимально детально.

Предложенный алгоритм состоит из следующих основных этапов:

1. Предобработка изображения
2. Извлечение особенностей изображения
3. Генерирование графа отношения особенностей(FRG)
4. Классификация

Далее будет подробно описан каждый этап алгоритма.

1. Предварительная обработка изображений

Для работы алгоритма необходимо предварительно обработать каждое входное изображение следующим образом:

* Разбить текст на строки. Пример представлен на рисунке



Рисунок 1. Сегментированный документ.

В нашей работе мы использовали базу данных[5], которая предоставляла уже заранее сегментированные тексты.

* Получив набор изображений строк, следует обработать каждую инвертировав ее и применив необходимые морфологические операторы для устранения нежелательных разрывностей и/или сгущений. Пример предварительно обработанной строки текста представлен на рисунке 2. Для обработки такого рода была использована библиотека OpenCV версии 2.4.9[8]. Библиотека предоставляет функционал для использования морфологических операторов.



Рисунок 2. Обработанная строка

1. Извлечение особенностей изображения

Для каждого входного изображения особенности извлекались с помощью двумерных фильтров Gabor[6], настроенных на разные ориентации и фильтров XGabor[4]. Более подробно об этих фильтрах речь пойдет далее.

* + Фильтр Gabor

 Фильтр Gabor широко применяется в задачах распознавания, так как является отличным инструментом для оптимальной фильтрации изображения в частотной и пространственной областях или, выражаясь менее формально, он в большей степени чувствителен к линиям определенной толщины и ориентации. Визуальные системы некоторых млекопитающих (кошки, макаки и так далее) могут быть аппроксимированы такими фильтрами.

Двухмерный фильтр Gabor (рисунок 5) задается произведением двумерной синусоиды (рисунок 3) с двумерной функцией Гаусса (рисунок 4).



Рисунок 3. Двумерная синусоида.



Рисунок 4. Двумерная функция Гаусса.



Рисунок 5. Двумерный фильтр Габора.

 В общем случае фильтр Габора задается равенством 1:

 (1)

Где u и v определены системой равенств 2:

 ; (2)

Переменная отвечает за длину волны двумерной синусоиды. Переменная отвечает за ориентацию паттернов на который настроен фильтр. Фильтр будет реагировать на паттерны, под углом . Переменная отвечает за смещение синусоиды по фазе. Переменная является стандартным отклонением функции Гаусса. отвечает за эллиптичность фильтра.

Пусть – функция, представляющая наш двумерный сигнал, за который мы принимаем входное изображение. Тогда реакцию фильтра, которую обозначим за , в точке можно задать равенством 3.

  (3)

Имея набор фильтров Gabor, настроенных на различные ориентации и с оптимально подобранными для входных текстов параметрами, можно добиться извлечения множества особенностей, хорошо характеризующих тот или иной почерк писателя.

При реализации системы для извлечения Gabor особенностей была использована библиотека OpenCV версии 2.4.9. Данная библиотека уже содержит оптимизированную реализацию вычисления ядра фильтра Gabor. Для входных данных были подобраны оптимальные параметры, которые передавались методу cv2.getGaborKernel(). Данные подбирались на основе уже замеченных соотношений. Наблюдения, представленные в [7], позволили значительно облегчить подбор параметров, устанавливая некоторые эмпирические соотношения между ними. Например, параметр

* + Фильтр XGabor

Данный фильтр является модификацией фильтра Gabor. Он был предложен в работе [4] как пригодное средство для детекции особых криволинейных паттернов.

Для определения фильтра XGabor необходимо определить следующую функцию (равенство 4):

 (4)

Важными параметрами являются . Они задают пропорцию длин сторон окаймляющего прямоугольника для кривой, которую требуется детектировать.

Тогда реакцию фильтра, которую обозначим за , в точке можно задать равенством 5.

 (5)

Получения ядра фильтра XGabor было реализовано вручную. Его реализация так же основывалась на уже существующих и протестированных методах библиотеки OpenCV.

В представленной работе каждая строка писателя обрабатывалась фильтрами Gabor и фильтрами XGabor. Обозначим за изображение, полученное после обработки изначального изображения фильтром c номером . За обозначим ширину изображения, за – высоту изображения. После обработки каждого изображения фильтрами для автора создавался вектор особенностей , где задавалось равенством 6, а равенством 7.

 (6)

 (7)

Далее вектор нормализовался. Таким образом, для писателя получался набор нормированных - мерных векторов, где .

Для более быстрой работы с векторными вычислениями была использована библиотека NumPy[9]. Данный модуль был специально разработан для языка программирования Python для быстрых и оптимизированных математических вычислений.

1. Конструирование FRG
* Сначала между извлеченными авторскими особенностями устанавливались отношения , ,,,. Для задания этих отношений вводились числовые переменные , ,,,.

Чтобы определить эти переменные определялись ещё две: и .

Пусть имеется некоторый вектор .

Положим

Тогда определим , ,,,равенствами (8) – (10).

 (8)

 (9)

 (10)

* После вычисления этих переменных для каждой пары особенностей необходимо задать пять отношений. Из-за наличия быстрых способов работы с n-мерными массивами, предоставляемых библиотекой NumPy был выбран матричный способ хранения и обработки данных. Для каждого вектора строилось 5 матриц , ,,,, где . В свою очередь, величины , ,,, определялись равенствами (11) – (15).

 (11)

 (12)

 (13)

 (14)

 (15)

Поведение величин , ,,, в зависимости от представлено на рисунке 6.

 

Рисунок 6. График поведения , ,,,.

Данные отношения характеризуют насколько особенности отличаются между собой. Как видно из графика сумма , ,,, для любой пары особенностей не превосходит 1.

* После вычисления вышеописанных матриц, характеризующих отношения между особенностями строятся пять финальных матриц , ,,,, где задаётся равенством 13.

 (16)

* Как только вычислены , ,,, генерируется ориентированный невзвешенный граф отношения особенностей , для каждого ребра которого справедливо утверждение 17.

(17)

Как видно из рисунка 6, сумма, стоящая в неравенстве, не превосходит 2, и если она больше единицы, то это означает, что особенность в текстах данного автора «как правило» численно превосходила особенность . В итоге, в графе существование ребра от к равносильно тому, что особенность в большинстве случаев численно превосходит особенность . Из этого следует, что получившийся граф не имеет циклов, иначе устройство графа в силу транзитивности неравенства оказалось противоречивым.

* Таким образом, каждому автору в конце данного этапа алгоритма поставлен в соответствие некоторый FRG, некоторым образом характеризующий авторский почерк. Полученный FRG в дальнейшем будет использоваться при классификации.
1. Классификация

При подаче тестового изображения, авторство которого необходимо установить, по алгоритму, представленному выше строится соответствующий изображению FRG, который мы далее будем обозначать за . Задача заключается в поиске графа наиболее «схожего» с . С этой целью для и каждого вычисленного на предыдущем этапе графа вычисляется некоторая величина , которая показывает степень схожести. Считая, что число известных в ходе работы авторов равно , наиболее схожий с граф схожим с , если для него будет выполнено утверждение 18.

(18)

Опишем процесс вычисления . Необходимо ввести такое понятие как высота вершины графа, которую мы будем обозначать за при . Величина будет обозначать максимальную длину пути от до какой-либо вершины, из которой не исходит ни одного ребра(такие вершины мы для краткости далее будем называть «листьями»). Очевидно, что для любого листа будет верно .

Также перед описанием алгоритма вычисления требуется ввести величину , которая показывает число одинаковых путей в графах и , берущих начало от вершины (если только имеется в обоих графах).

Как вычислить ? Определим множество «соседей» вершины в графе равенством 16.

 (19)

Так как мы работаем с ориентированными графами, не имеющими циклов, верно равенство 20.

 (20)

Теперь можно описать сам алгоритм вычисления :

* Для всех вершин вычислить .
* Отсортировать все ребра , считая, что
* Получив отсортированный массив ребер, обработать его с первого элемента, посчитав по формуле 17.
* теперь получается суммированием вычисленных величин .

Вышеописанный алгоритм классификации был реализован и успешно протестирован вручную. Для построения графов и быстрых операций с ними была выбрана для использования библиотека NetworkX[10]. Этот модуль предоставляет один из наиболее обширных функционалов по работе с графами различного типа для языка Python.

* 1. Тестирование алгоритма
		1. Тренировочные данные

В качестве исходных данных для работы системы была взята база данных **K**FUPM **H**andwritten **A**rabic **T**ex**T** (KHAT, версия 1, дата выпуска: сентябрь 2012) [5]. В формировании содержимого базы данных участвовало 1000 человек из разных стран, различного пола, возраста. База содержала изображения различных текстов. Важной особенностью базы являлось то, что она содержала заранее сегментированные тексты.

* + 1. Эксперименты

Был проведен ряд экспериментов, чтобы проверить процесс классификации, описанный выше, на арабографических рукописных документах. Каждый эксперимент задавался следующими параметрами:

* - Количество авторов, участвующих в эксперименте
* – отношение числа тренировочных изображений к количеству тестовых изображений
* - число извлекаемых особенностей

Точность для каждого из экспериментов обозначена как. Эта величина показывает процент правильно классифицированных документов. Эксперименты были проведены следующим образом:

* Для первой серии экспериментов было использовано около 4000 файлов – около 4 файлов для каждого писателя. Файлы были сегментированы вручную, так как здесь мы не решали проблему сегментации. Каждый файл содержал от 2 до 5 сегментированных строк текста. Авторы были выбраны случайным образом. Их число было равно 3. Для каждого автора было отобрано более 12 строк текста. был установлен в 8/4 и 4/8. был установлен в 8 и 24.
* Вторая серия экспериментов задействовала 13 тысяч строк текста различных авторов. В данном эксперименте сегментация производилась не вручную, а использовались заранее сегментированные тексты, которые предоставляла база данных KHATT. Для каждого писателя было выбрано более 12 строк текста. Число писателей было положено равным 10.
* Третья и четвертая серии экспериментов основывались на второй серии, но число авторов и число извлекаемых особенностей было увеличено.

Наиболее показательные результаты экспериментов представлены в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 3 | 4/8 | 8 | 66 |
| 3 | 8/4 | 8 | 100 |
| 3 | 4/8 | 24 | 100 |
| 3 | 8/4 | 24 | 100 |
| 10 | 4/8 | 8 | 50 |
| 10 | 8/4 | 8 | 70 |
| 10 | 4/8 | 24 | 80 |
| 10 | 8/4 | 24 | 80 |
| 15 | 4/8 | 24 | 40 |
| 15 | 8/4 | 24 | 45 |
| 15 | 4/8 | 36 | 40 |
| 15 | 8/4 | 36 | 50 |
| 20 | 8/4 | 36 | 45 |
| 30 | 8/8 | 36 | 40 |

Таблица 1. Тестирование алгоритма на арабографических документах

Как показали эксперименты, самыми значимыми оказались особенности, извлекаемые с помощью фильтра Gabor. В ряду экспериментов увеличение числа извлекаемых особенностей только ухудшало результат классификации.

Видно, что алгоритм упоминалось выше, может быть успешно использован для арабских текстов при небольшом количестве авторов, так как точность ощутимо падает при росте их числа. Точность алгоритма увеличивается с увеличением количества данных обучения.

* 1. Применение алгоритма при кластеризации документов

Процесс кластеризации известен как процесс группирования входных

данных по группам таким образом, чтобы все элементы внутри каждой группы

были похожи друг на друга в определенном смысле. Рассмотрим набор

арабских рукописных текстовых документов . Наша задача заключается в разделении в наборы так, что каждый содержит как можно больше изображений

определенного писателя и как можно меньше изображений иных авторов. - обычно неизвестный параметр, который должен быть вычислен.

Задача кластеризации может быть сведена к минимизации специальной функции стоимости. Вычисление этой функции основано на расчете расстояния между элементами кластеров. Таким образом, возникает необходимость определения некоторой функции расстояния между двумя элементами. Предположим, что необходимо вычислить расстояние между двумя изображениями и . Пусть и - строки текста, которые могут быть извлечены из и соответственно. Пусть и - FRG графы, которые получаются из и соответственно по алгоритму, описанному ранее. Определим расстояние между и равенством 21.

 (21)

Как видно, чем больше схожи графы, тем меньше величина , а, следовательно, и «ближе» два изображения.

После того, как расстояние между двумя отсканированными текстами определено, мы можем произвести кластеризацию. Мы протестировали процесс кластеризации, используя следующие алгоритмы: K-means[11], Online K-Means[12], РАМ[13] и DBSCAN[14]. Каждый из этих алгоритмов ввиду специфичности кластеризуемых данных был реализован вручную и адаптирован под FRG.

Для оценки результатов работы алгоритмов мы определили свою функцию стоимости разбиения на кластеры, алгоритм вычисления которой приведен ниже.

Алгоритм вычисления :

* + Для каждого автора
		- Выбираются «**материнские кластеры**» – кластеры **наименьшего размера**

среди тех, что содержат **наибольшее количество рукописей автора.**

* + - Ищется кластер **без хозяина**.
			* айден:

 помечается как хозяин .

 = (\*)

* + - * **не** найден:
				+ Среди материнских кластеров ищется , чей обладатель имеет наменьшее .
				+ – другой автор.
				+ пересчитывается в соответствии с (\*).

На рисунке 7 представлен пример разбиения входного множества документов на четыре кластера. Каждый документ схематично помечен номером и цветом, которые обозначают автора. Тогда для разбиения, представленного, на рисунке 7 значение Cost функции будет равно 0.36.



Рисунок 7. Пример разбиения входного множества документов на кластеры.

Был проведен ряд экспериментов для каждого алгоритма кластеризации, упомянутого выше. Каждый эксперимент определялся следующими параметрами:

* – число авторов, участвующих в эксперименте (целевое число кластеров)
* – число извлекаемых особенностей

Параметр был равен в алгоритмах - means, PAM. Online k-Means и

DBSCAN определяют во время работы. Во время тестов алгоритма DBSCAN несколько дополнительных параметров должны были быть заданы. Алгоритм DBSCAN параметризуется двумя величинами, которые обозначаются как и . Во время экспериментов был определен на основе максимального сходства между графами, а параметр был взят из диапазона . База данных KHAT была взята как источник отсканированных арабографических рукописных документов. Во время экспериментов параметр был приравнен к 3, 10 и 20. Число извлекаемых особенностей было положено равным 15, 32, 64. Каждый писатель во время экспериментов имел в среднем 12 строк текста. Для алгоритма online k-means все тексты обрабатывались в произвольном порядке. Кластеризация производилась несколько раз. Результаты экспериментов с представлены в таблице 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Authors | Features | K-Means | Online K-Means | PAM | DBSCAN |
| 3 | 15 | 33 | 33 | 60 | 33 |
| 3 | 32 | 66 | 33 | 60 | 33 |
| 3 | 64 | 66 | 50 | 66 | 66 |
| 10 | 15 | 10 | 10 | 55 | 33 |
| 10 | 32 | 20 | 20 | 65 | 40 |
| 10 | 64 | 20 | 25 | 65 | 55 |
| 20 | 15 | 5 | 5 | 45 | 12 |
| 20 | 32 | 8 | 5 | 53 | 20 |
| 20 | 64 | 10 | 8 | 55 | 25 |

Таблица 2. Результаты экспериментов с кластеризацией

На трех кластерах алгоритм k-means показал 99 процентную точность. Как видно, точность кластеризации сильно зависит от числа извлекаемых особенностей.Online

k-means и DBSCAN показали приличные результаты, учитывая, что параметр не использовался в процессе кластеризации.

1. **Заключение**

В результате проделанной работы была реализована система для автоматической идентификации автора арабографического рукописного документа по почерку его создателя. Система использовала алгоритм, в основе которого лежал алгоритм, предложенный для идентификации автора персидских рукописей, который в своей работе[5] описали B. Helli и E. Moghaddam. Система была реализована на языке программирования Python с использованием оптимизированных библиотек. Была проведена настройка системы для работы с арабографическими рукописными документами. В качестве данных для тестирования была взята база данных KHAT, которая содержала заранее сегментированные на строки рукописи тысячи авторов. Были проведен ряд тестов для анализа работы алгоритма для решения задачи классификации арабографических рукописных документов. Алгоритм показал достойные результаты на арабографических рукописных текстах при наличии достаточного количества тренировочных данных и небольшого числа авторов. Также в систему была добавлена функциональность кластеризации входного множества документов с использованием алгоритмов кластеризации K-Means, Online K-Means, PAM, DBSCAN. Был произведен ряд экспериментов для установления точности используемого алгоритма при решении задачи кластеризации. Эксперименты показали, что алгоритм может быть пригоден для использования при кластеризации документов лишь небольшого числаавторовс использованиемалгоритмов кластеризации PAM.

Так как при числе авторов, превышающем 20, точность алгоритма значительно ухудшалась, не было проведено тестов с большим объемом данных. Поэтому в качестве возможных улучшений точности реализованной системы можно рассматривать:

* Использование особенностей другого типа
* Использование большего числа данных для каждого автора
1. **Список литературы**

[1] H.E.S. Said, T.N. Tan, K.D. Baker “Personal identification based on

Handwriting”, Pattern Recognition, Vol. 33(1), 2000.

[2] Imran Siddiqi , Nicole Vincent “Text independent writer recognition using

redundant writing patterns with contour-based orientation and curvature

features”, Pattern Recognition, Vol 43(11), 2010.

[3] Sameh M. Awaida, Sabri A. Mahmoud “Writer identification of Arabic

text using statistical and structural features”, Cybernetics and Systems:

An International Journal, Vol. 44(1), 2013.

[4] B.Helli, M.E. Moghaddam “A text-independent Persian writer identification

based on feature relation graph (FRG)”, Pattern Recognition, 2010.

[5] S. A. Mahmoud, I. Ahmad, M. Alshayeb, W. G. Al-Khatib, M. T.

Parvez, G. A. Fink, V. Margner, and H. EL Abed KHATT: Arabic Offline

Handwritten Text Database, In Proceedings of the 13th International

Conference on Frontiers in Handwriting Recognition (ICFHR 2012),

Bari, Italy, 2012, pp. 447-452, IEEE Computer Society.

[6] V. Shiv Naga Prasad and Justin Domke “Gabor filter visualization”, Technical Report, University of Maryland, 2005.

[7] P. Fuller “Gabor filter for scientists and engineers”, URL:

patrick-fuller.com/gabor-filter-image-processing-for-scientists-and-engineers-part-6/

[8] OpenCV library, URL: opencv.org

[9] NumPy library, URL: numpy.org

[10] NetworkX library, URL: networkx.github.io

[11] K. Teknomo “K-Means Clustering Tutorials”, 2007, URL: people.revoledu.com\kardi\ tutorial\kMean\

[12] A. King “Online k-Means Clustering of Nonstationary Data”, 2012

[13] 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

[14] Martin Ester, Hans-Peter Kriegel, Jrg Sander, Xiaowei Xu “A Density-

Based Algorithm for Discovering Clusters in Large Spatial Databases

with Noise”, AAAI Press, 1996.