УДК 004.021

Е.А. Кленина

ИССЛЕДОВАНИЕ ПРОБЛЕМЫ КЛАСТЕРИЗАЦИИ ДАННЫХ САЙТОВ С ОТКРЫТЫМИ API

Проведено исследование проблемы кластеризации данных сайтов с открытыми API. Выполнен выбор метода кластеризации, показывающего наилучшие результаты на эталонных данных. Произведена модификация данного метода с использованием различных мер сходства. Исследованы основные принципы работы с API сайтов и примеры кластеризации их данных.

Ключевые слова: *API, кластеризация, мера сходства.*

Введение. С появлением сети Интернет возможность собирать информацию от тысяч и даже миллионов людей открыла широчайший спектр новых возможностей по получению новых знаний, исходя из данных от независимых респондентов. Кластеризация данных сайтов позволяет вырабатывать маркетинговые рекомендации пользователям в соответствии с их предпочтениями, упрощать доступ к информации группированием ее по тематикам и многое другое. Объем информации, получаемой от респондентов в сети Интернет, очень велик и только все больше увеличивается, что требует эффективной автоматизации кластерного анализа.

С точки зрения рекламодателя, маркетолога или социолога, очень интересно проводить анализ информации социальных сетей и интернет-магазинов и аукционов.

Цель работы – исследование проблемы кластеризации данных сайтов с открытыми API, автоматизация процесса получения данных такого рода с последующим обнаружением среди них групп схожих объектов.

Теоретическая часть. Задача кластеризации состоит в разделении исследуемого множества объектов на группы «похожих» объектов, называемые кластерами. Решением задачи является отнесение каждого из объектов данных к одному (или нескольким) из заранее неопределенных классов [1].

Кластерный анализ включает следующие этапы.

1. Подбор данных для кластеризации. Подразумевается, что имеет смысл кластеризовать только количественные данные.

2. Выбор тех метрик данных, по которым будут группироваться объекты данных.

3. Вычисление значений той или иной меры сходства (или различия) между объектами.

4. Выбор метода кластерного анализа и его применение для обнаружения групп похожих объектов.

5. Проверка достоверности результатов кластерного решения.

Существует несколько классификаций методов кластерного анализа. По одной из них все методы разбиения на кластеры можно подразделить на иерархические и неиерархические [1].

При использовании иерархических алгоритмов необходимо заранее определить большое число параметров, определяющих их работу и условия остановки. В случаях недостаточности информации на начальном этапе это затруднительно. Однако в результате их работы удается однозначно определить число кластеров.

Неиерархические алгоритмы пытаются сгруппировать данные (в кластеры) таким образом, чтобы целевая функция алгоритма разбиения достигала экстремума (минимума) [1].

К неиерархическим алгоритмам кластеризации относятся алгоритмы k-means, c-means и др.

По другой классификации алгоритмы кластеризации делятся на четкие (не использующие нечеткую логику) и нечеткие (соответственно использующие аппарат нечетких отношений).

Недостатками методов, не использующих нечеткую логику, являются использование для анализа центров кластеров, а не всех данных, и отнесение объектов только к одному из кластеров, а не его частичное распределение по классам.

Алгоритм Fuzzy C-Means и алгоритм кластеризации по Гюстафсону-Кесселю устраняют второй недостаток, но накладывают ограничение на форму кластеров.

Перечисленные недостатки устранены в кластеризации данных с помощью аппарата нечетких отношений.

Идея кластеризации с помощью нечетких отношений состоит в следующем. Свойства реф-

лексивности, симметричности и транзитивности обобщаются на их нечеткие аналоги для получения аналога классов эквивалентности в теории нечетких отношений.

Сравнение образцов данных осуществляется с помощью отношения α-квазиэквивалентности и шкалы α-квазиэквивалентности.

В рамках решения задачи кластеризации данных с помощью нечетких отношений необходимо исследовать алгоритм построения шкалы отношения α-квазиэквивалентности и разработать его реализацию, а также разработать алгоритм построения кластеров по полученной шкале α-квазиэквивалентности.

Этап 1 – построение шкалы α-квазиэквивалентности. Дано: множество образцов данных $X = \{x_i\}_{i=1}^{Q}$, где $x_i = (x_{i1},...,x_{in})$, $x_{ij} \in R$, n – размерность образцов данных, $Q = |X|$ – мощность множества X.

Схема работы метода кластеризации с помощью нечеткой логики показана на рисунке 1.

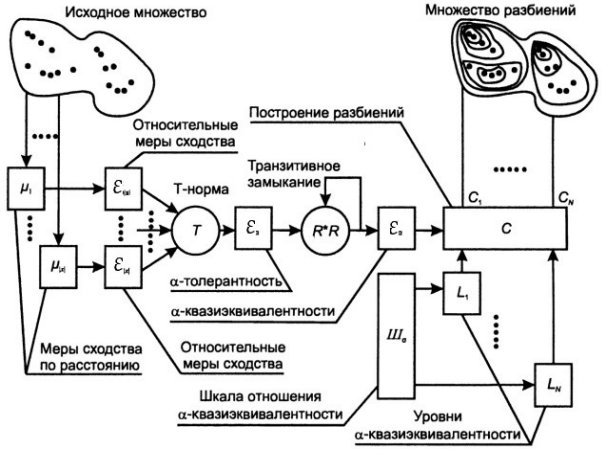

Рисунок 1 – Схема работы алгоритма α-квазиэквивалентности

Получение шкалы α-квазиэквивалентности.

1. Построить для каждого образца данных $x_i = (x_{i1},...,x_{in})$, $x_{ij} \in R$ нормальную меру сходства по формуле:

$$\mu_{x_q}(x_i) = 1 - \frac{d(x_q, x_i)}{\max\limits_{k \in [1,Q]} (d(x_q, x_k))}, \; q, i = \overline{1, Q},$$

где d – расстояние по Евклиду.

2. Построить относительную меру сходства для пар образцов данных:

$$\varepsilon_{x_q}(x_i, x_j) = 1 - \left| \mu_{x_q}(x_i) - \mu_{x_q}(x_j) \right|, \; i, j, q = \overline{1, Q}.$$

3. Построить меру сходства образцов данных на множестве X по формуле:

$$\varepsilon(a,b) = T(\varepsilon_{x_1}(a,b),..., \varepsilon_{x_q}(a,b)) =$$

$$= \min_{i=\overline{1,Q}} \varepsilon_{x_i}(a,b), a,b \in X.$$

4. Построить транзитивное замыкание отношения меры сходства образцов данных на множестве X

$$R_\varepsilon^{|X|} = R_\varepsilon^{Q}, \; r_{ij}^{|X|} = r_{ij}^{Q}$$

$$R_\varepsilon^{q} = R_\varepsilon^{q-1} \circ R_\varepsilon.$$

В соответствии с определением операции композиции нечетких отношений

$$r_{ij}^{2} = S_{k=1}^{Q} T(r_{ik}, r_{kj}).$$

Построенное отношение $R_\varepsilon^{|X|}$ есть отношение α-квазиэквивалентности.

Шкала α-квазиэквивалентности может быть получена как множество различных элементов отношения, упорядоченных по возрастанию.

Построение кластеров по шкале α-квазиэквивалентности реализуется следующим образом. Каждому уровню шкалы соответствует отношение эквивалентности. Выбор каждого последующего уровня α-квазиэквивалентности порождает более детальное разбиение множества X.

Помещаем первый объект в первый класс. Далее в цикле выбираем все последующие объекты для определения их класса. Для каждого объекта проверяем его на принадлежность каждому из предшествующих классов. Если размещаемый элемент с каким-либо из размещенных элементов кластера имеет степень принадлежности отношению α-квазиэквивалентности менее уровня α, он должен быть размещен в другом кластере.

Кроме исследования вопросов кластеризации необходимо было также разработать алгоритм получения данных из открытых API сайтов.

Открытые API сайты – это, как правило, определенный набор HTTP-методов и определенные структуры HTTP-ответов в формате XML (тип SOAP) или JSON (тип REST). В связи с переходом на Web 2.0 имеется тенденция использовать REST вместо SOAP типа коммуникации, поэтому, когда результатом обращения к API является XML, лучше привести его к JSON для большей универсальности выходного формата данных.

После изучения работы с API сайтов EBay, ВКонтакте, Yahoo Finance были выявлены закономерности, которые позволили разработать схему работы с открытыми API (рисунок 2), которая затем была реализована в программном средстве кластеризации данных с помощью открытых API.

Рисунок 2 – Принцип получения данных через открытые API сайты

На рисунке 2 блок номер 1 соответствует прохождению регистрации разработчика на сайте документации к определенному API. Данная процедура индивидуальна для каждого сайта, поэтому блок 1 не поддается автоматизации.

С учетом возможностей современных программных платформ – работать с HTTP-запросами как в формате JSON, так и в XML, процессы формирования заголовков, отправки и преобразования форматов возможно автоматизировать. От пользователя понадобится только выбрать формат, ввести базовый url доступа к методу API и параметры запроса. Два последних параметра можно задать, найдя нужную информацию по методу в документации на сайте определенного API.

Экспериментальные исследования. Для выбора оптимального метода кластеризации необходимо провести сравнение результатов работы нескольких известных алгоритмов кластеризации. В данной работе первым для сравнения выбран один из алгоритмов без нечеткой логики – алгоритм классической кластеризации Хартигана и Вонга (алгоритм k-средних).

Недостатками методов без нечеткой логики являются использование для анализа центров кластеров, а не всех данных и отнесение объектов только к одному из кластеров, а не его частичное распределение по классам. Поэтому для сравнения также взят алгоритм с нечеткой логикой Fuzzy C-Means. Его недостатком является ограничение на форму кластеров. Перечисленные недостатки устранены в кластеризации данных с помощью аппарата нечетких отношений. Поэтому алгоритм кластеризации с помощью нечетких отношений также вошел в число сравниваемых алгоритмов.

В отличие от других алгоритмов, кластеризация с помощью нечетких отношений не требует априорного задания числа кластеров, которое можно определить в конце решения исходя из приемлемого значения уровня α-квазиэквивалентности. Обычно таким значением является 0,75.

Сравнивать 3 данных алгоритма и делать выводы о больших преимуществах одного из них для решения без количественной оценки не представляется возможным. Такой оценкой может служить F-мера, описанная далее, но необходимо вводить эталонные разбиения на кластеры. В решении задачи кластеризации данных сайтов с открытыми API при отсутствии экспертных мнений невозможно определить эталонные разбиения на кластеры. Поэтому было принято решение сгенерировать тестовые эталонные кластеры, используя равномерное распределение, и подать их на вход всех трех алгоритмов, чтобы выделить алгоритм с наибольшим приоритетом для кластеризации данных в кластеры неизвестной структуры.

Построение эталонных кластеров, каждый из которых содержал бы n-равномерно распределенных точек, было реализовано в 3 этапа.

Был выбран сегмент двумерного пространства с координатами от [0; 500] и разделен по оси X и Y на квадранты по числу кластеров k. В сегменте случайным образом по равномерному распределению были размещены квадраты, соответствующие каждому кластеру, по размеру 1 квадранта.

Для каждого квадрата пошагово квадрантами увеличивался его размер до тех пор, пока он не начинал пересекаться с другими. Пересечение не допускалось, поскольку случай пересекающихся кластеров частный и очень сложен для базовых алгоритмов кластеризации. В конце этапа были получены прямоугольники, случайно размещенные в сегменте пространства.

На данном этапе нужно было усложнить форму кластеров, вписанных в прямоугольники, полученные на втором этапе. Если случайная величина принимает любое значение из некоторого множества с одинаковой вероятностью, то любое значение из произвольного подмножества этого множества также равновероятно. Тогда можно строить кластеры нетривиальной формы с равномерным распределением объектов, вырезая их из объемлющего n-мерного параллелепипеда, точки в котором генерируются случайным образом по равномерному распределению. Таким образом, равномерно распределенные точки

генерировались для всего сегмента [0;500], а затем уже в зависимости от попадания в сгенерированные прямоугольники или вписанные в них формы кластеров были отнесены к определенным кластерам.

Нетривиальные формы кластеров были сформированы проверкой попадания точки в уравнение окружности:

$$x^2 + y^2 = r^2.$$

Два таких уравнения позволили строить кластеры кольцевых форм, а три – вложенные кластеры.

Поскольку центры окружностей должны были находиться уже не в точке (0;0), а в полученных на этапе 2 прямоугольниках, было использовано преобразование параллельного переноса.

Для того чтобы создавать кластеры в форме эллипсов к кластерам в форме окружности было применено преобразование масштабирования.

Исходными данными для генерирования эталонных кластеров был выбран фрагмент двумерного декартового пространства [0;500]: достаточное по величине число кластеризуемых объектов – 1000, число кластеров – 10. Уровень сложности расположения кластеров был выбран таким образом, что допускается наличие вложенных кластеров, близко расположенных кластером, но не пересекающихся друг с другом.

Один из программно сгенерированных наборов эталонных кластеров для сравнения трех алгоритмов кластеризации показан на рисунке 3.

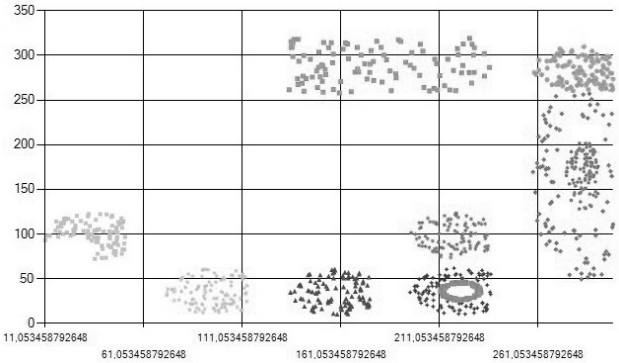

Рисунок 3 – Пример двумерных тестовых данных для сравнения алгоритмов

Метрикой, позволяющей оценить качество обнаружения кластеров похожих объектов каждым из алгоритмов, может служить F-мера. Введем для этого необходимые определения.

Точность определяется как отношение числа правильно отнесенных к кластеру объектов к общему числу объектов в эталонном кластере:

$$P(C_i, M_j) = \frac{n_{ij}}{n_j},$$

где $n_j = |M_j|$ – число элементов в кластере M_j (эталонное разбиение), C_i – кластеры, полученные в результате работы алгоритмов; $n_{ij} = |M_j \cap C_i|$ – число общих элементов M_j и C_i.

Полнота – отношение числа правильно отнесенных к кластеру объектов к общему числу объектов в полученном кластере:

$$R(C_i, M_j) = \frac{n_{ij}}{n_i},$$

где $n_i = |C_i|$ – число элементов в кластере C_i (вычисленное разбиение), M_j – кластеры эталонного разбиения, $n_{ij} = |M_j \cap C_i|$ – число общих элементов M_j и C_i.

Иногда бывает полезно объединить точность и полноту в одной усреднённой величине. Для этой цели среднее арифметическое не подходит, так как, например, алгоритму достаточно поместить все объекты в каждый кластер, чтобы обеспечить равную единице полноту при близкой к нулю точности, и среднее арифметическое точности и полноты будет не меньше 1/2. Среднее гармоническое не обладает этим недостатком, поскольку при большом отличии усредняемых значений приближается к минимальному из них.

Поэтому хорошей мерой для совместной оценки точности и полноты является F-мера, которая определяется как взвешенное гармоническое среднее точности P и полноты R:

$$F(C_i, M_j) = \frac{2 * P(C_i, M_j) * R(C_i, M_j)}{P(C_i, M_j) + R(C_i, M_j)} = \frac{2 * n_{ij}}{n_i + n_j}.$$

F-мера принимает значения в интервале [0;1]. Чем значение ближе к 1, тем качественнее проведена кластеризация.

Для эталонных кластеров на рисунке 3 был выполнен прогон каждого из трех реализованных алгоритмов и программный расчет F-мер рассчитанных разбиений по сравнению с эталонным и были получены результаты, представленные в таблице 1.

Таблица 1 – Результаты сравнения алгоритмов K-Means, C-Means, алгоритма α-квазиэквивалентности

	F-мера	Время работы, с
K-Means	0,76	0,06
C-Means	0,83	0,45
Алгоритм α-квазиэквивалентности	0,92	180,72

Как видно, по результатам сравнения под-

твержда́ются сведения из литературных источников [1] о том, что алгоритм α-квазиэквивалентности обнаруживает кластеры более универсальных форм, но данный алгоритм показывает плохое время работы на большой выборке данных – 1000 объектов за 180 с, поэтому в реализации программы кластеризации данных сайтов с открытыми API будут использованы все три алгоритма, но большим приоритетом будет обладать алгоритм α-квазиэквивалентности из-за близкого к 1 значения F-меры и отсутствия необходимости в априорном задании числа кластеров.

Один из вариантов проверки качества кластеризации – использование нескольких методов и сравнение полученных результатов. Отсутствие подобия не будет означать некорректность результатов, но присутствие похожих групп считается признаком качественной кластеризации [2]. Таким образом, несмотря на вывод о наибольшей пригодности одного из алгоритмов для решения задачи, в программной реализации инструмента для автоматизированной кластеризации данных сайтов с открытыми API следует использовать все три алгоритма для первичной оценки качества результатов друг друга. В пользу этого говорит еще и тот факт, что алгоритм k-средних, в отличие от двух других, имеет формальную оценку качества кластеризации. По тому, насколько сильно средние значения для каждого кластера для всех измерений или хотя бы большей их части отличаются друг от друга, можно судить о качестве кластеризации. Признаком хорошей кластеризации будет считаться сильное отличие данных значений.

Далее была проведена модификация алгоритма α-квазиэквивалентности – добавлена нормализация исходных данных и применение различных мер сходства.

В модификации было применено нормирование одним из известных способов – делением исходных данных на среднеквадратичное отклонение соответствующих переменных.

$$z_{ij} = \frac{a_{ij} - \overline{a}_j}{\sigma_j}.$$

Обозначения:

z_{ij} – нормированное значение элемента a_{ij} матрицы исходных данных;

$\overline{a}_j$ – среднее значение элементов по j–му столбцу матрицы исходных данных (другими словами – среднее значение измеренного признака X_j по всем подвергнутым измерению объектам);

σ_j – среднеквадратичное отклонение, вычисленное по j–му столбцу матрицы исходных данных (среднеквадратичное отклонение значений признака X_j).

При модификации алгоритма исследуется применение следующих мер сходства.

Манхэттенское расстояние. Представляет сумму длин отрезков между координатами объектов по осям координат. Часто применение этой меры дает такие же результаты, как Евклидово расстояние. Однако для этой меры влияние больших разностей уменьшается (так как они не возводятся в квадрат):

$$d(x_q, x_k) = \sum_i \left| x_{qi} - x_{ki} \right|.$$

Степенное расстояние. Позволяет уменьшать или увеличивать влияние одной из метрик, значения которой имеют сильное различие. Степенное расстояние вычисляется по следующей формуле:

$$d(x_q, x_k) = \sqrt[r]{\sum_i^n (x_{qi} - x_{ki})^p},$$

где r и p – параметры, определяемые пользователем.

Параметр p ответственен за постепенное взвешивание разностей по отдельным координатам, параметр r ответственен за прогрессивное взвешивание больших расстояний между объектами. Когда r=2 и p=2, степенное расстояние эквивалентно расстоянию Евклида.

Коэффициент корреляции Пирсона. Коэффициент корреляции Пирсона r, который является безразмерным индексом в интервале от 0 до 1,0 включительно, отражает степень линейной зависимости между двумя множествами данных [3]. Коэффициент корреляции Пирсона равен 1, если вектора данных полностью совпадают, а близок 0, если между векторами нет связи:

$$r = \frac{n \cdot \sum_{i=1}^n (x_i y_i) - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\left[n \cdot \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2\right] * \left[n \cdot \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2\right]}},$$

где n – количество значений в векторе.

Коэффициент корреляции Пирсона позволяет хорошо скорректировать близость векторов данных для кластеризации, когда они отражают присутствие или отсутствие элементов и количество элементов для каждого вектора серьезно разнится [3].

Коэффициент Танимото. Если вектора данных содержат только 0 и 1 как с примером данных сайта «ВКонтакте», которые означают отсутствие и присутствие каких-то признаков, полезно определять перекрытие между векторами, имеющими общие признаки. Коэффициент Та-

нимото – отношение мощности пересечения множеств к мощности их объединения. Коэффициент, равный 1, означает, что вектора полностью перекрываются, а 0 – что в них нет ни одного одинакового элемента.

Первые две меры сходства заменяют Евклидово расстояние на 1 этапе алгоритма кластеризации с помощью нечетких отношений – вычисления нормальной меры сходства по формуле:

$$\mu_{x_q.}(x_i) = 1 - \frac{d(x_q, x_i)}{\max\limits_{k \in [1,Q]} (d(x_q, x_k))}, \; q, i = \overline{1, Q}.$$

А вторые две меры полностью заменяют нормальную меру сходства. Данный вывод был сделан на основании определения меры сходства и нормальной меры сходства. Мера сходства по расстоянию – функция $\mu_{x_0}(x_0) : X -> [0,1]$, $x_0 \in X$ линейно убывающая по расстоянию и принимающая значение $\mu_{x_0}(x_0) = 1$. Нормальной мерой сходства по расстоянию с образцом данных называется такая мера, которая принимает свои граничные значения на множестве X [1]. Коэффициент корреляции Пирсона и коэффициент Танимото принимают значения [0,1] на множестве X и тем ближе к 1, чем больше расстояние между образцами, а ближе к 0, чем меньше расстояние, то есть убывают по расстоянию.

Блок-схема модифицированного фрагмента вычисления нормальной меры сходства для алгоритма кластеризации на основе нечетких отношений представлена на рисунке 4.

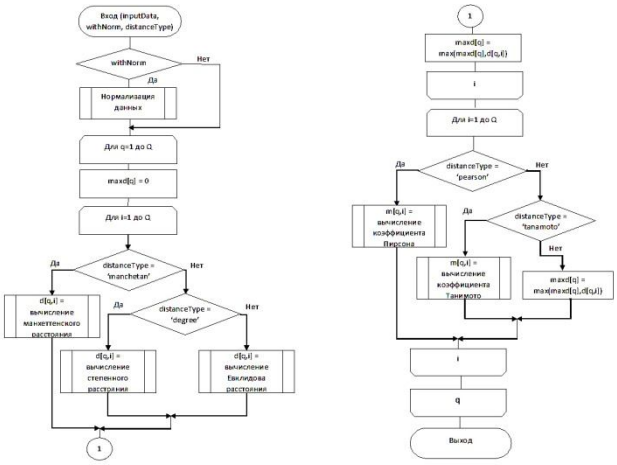

Рисунок 4 – Модифицированный фрагмент вычисления нормальной меры сходства алгоритма α-квазиэквивалентности

Приведем результаты экспериментов того, как алгоритм кластеризации на основе нечетких отношений и его модификация могут быть применены к данным сайтов с открытыми API.

Пример 1. Кластеризация проводилась для товаров онлайн-аукциона eBay, полученных с помощью API-метода findItemsAdvanced с фильтром keyword=leather&size=L&gender=man по ценам и проценту положительных отзывов продавцу товара. Процент получен вызовом для каждого товара метода getSingleItem.

Таблица 2 – Данные о товарах с сайта eBay

2618	4009	2519	1314	2520	2521	2522
(148; 99)	(2; 96)	(104; 98)	(2; 99)	(75; 100)	(104; 100)	(119; 98)

В результате кластерного анализа данных сайта eBay (методом α-квазиэквивалентности без нормализации и с мерой сходства – Евклидово расстояние программа выдала в результате разбиение – {2618}, {4009, 1314}, {2519, 2521, 2522}, {2520}. Уровень α-квазиэквивалентности выбирается минимально достаточным ≥0,75.

Результаты можно интерпретировать следующим образом – выделилась группа товаров с очень низкой ценой, один товар с умеренной ценой, товары с высокой ценой, один товар с очень высокой ценой.

После включения нормализации изменилось и количество кластеров, и их состав. Объекты еще четче разделились по числовым значениям своей первой метрики (ось X). Из этого следует вывод, что метрика, имеющая больший разброс по значениям, так и оставила за собой самый большой вклад в результат кластеризации.

Полученные с применением манхеттенского расстояния в алгоритме α-квазиэквивалентности без нормализации результаты кластеризации полностью совпадают с результатами применения Евклидова расстояния.

С использованием степенного расстояния удалось скорректировать степень влияния значений первой метрики (ось X). При r=1 и p=2 удалось получить новый вариант разбиения с тремя кластерами.

Коэффициенты корреляции Пирсона и Танимото применяются тогда, когда входные данные представлены псевдобулевыми значениями 0 и 1, означающими соответственно присутствие или отсутствие признака, поэтому не будут использованы в данном примере с числовыми входными данными.

При применении к данным примера 1 кластеризации алгоритмом k-means были получены результаты, немного отличающиеся от результатов алгоритма α-квазиэквивалентности. Количество кластеров выбрано таким же, как получаемое в результате работы алгоритма α-квазиэквивалентности, потому что на этот счет не имеется априорных данных.

$$K = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccccccccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

Алгоритм C-Means для примера 1, количества кластеров равного 4 и меры сходства – Евклидова расстояния дает такой же результат как алгоритм α-квазиэквивалентности.

Выводы по примеру 1: использование алгоритма α-квазиэквивалентности показывает адекватные результаты, в некоторой степени похожие на результаты работы двух других алгоритмов. Эффективной мерой влияния на результаты кластеризации показало себя степенное расстояние, которое может своим применением заменять и нормализацию входных данных.

Пример 2. Данные, полученные через API сайта ВКонтакте.

Входными данными выступает матрица интересов 30 пользователей,
где

$$k_{ij} = \begin{cases} 1, \textit{если на странице пользователя указан интерес} \\ 0, \textit{если на странице нет определенного интереса} \end{cases}.$$

По столбцам матрицы под номерами зашифрованы пользователи социальной сети, а по строкам их интересы:

1 – иностранные языки;
2 – фотография;
3 – путешествия;
4 – психология;
5 – автомобили;
6 – спорт;
7 – кино.

После кластеризации таких данных можно найти людей, чьи интересы схожи и рекомендовать им для общения друг друга.

Наиболее значимые экспериментальные результаты работы алгоритмов на данных примера 2 показаны в таблице 3.

Сравнив результаты в таблице и матрицу входных значений, можно логическим путем сделать вывод об адекватности каждого из разбиений. Однако при работе алгоритма α-квазиэквивалентности с коэффициентом Танимото получилось значительно большее число кластеров. Отладка алгоритма показала, что и уровней α-квазиэквивалентности получилось на 3-4 меньше, чем в других примерах. Из этого был сделан вывод, что коэффициент Танимото не очень эффективно применим к данному примеру. Выбор между алгоритмом k-means или α-квазиэквивалентности с коэффициентом Пирсона остается за пользователем, так как оба алгоритма дают адекватные результаты.

Таблица 3 – Результаты кластеризации данных о интересах пользователей ВКонтакте

Алгоритм (и мера сходства)	Разбиение множества пользователей
k-means	{1, 6, 8, 12, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27}, {2, 13, 14, 15, 17}, {3, 5, 9, 11}, {4, 7, 10}
α-квазиэквивалентности с коэффициентом Пирсона	{1, 6, 8, 12, 16, 18, 21, 26}, {2, 7, 14, 20}, {3, 4, 5, 10, 15, 22, 24, 25}, {9, 11, 13, 17, 18, 19, 23, 27}
α-квазиэквивалентности с коэффициентом Танимото	{1, 6, 8 21 26}, {2, 13, 14, 15, 17}, {3, 5, 9}, {4}, {7, 10, 20} {18} {19, 13, 27} {22, 24, 25}

Заключение. В статье проведено сравнение трех алгоритмов кластеризации. Показана эффективность алгоритма α-квазиэквивалентности на произвольных эталонных данных. Разработана модификация данного алгоритма и доказана на примерах целесообразность ее применения к данным сайтов с открытыми API.

Библиографический список

1. *Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И.* Методы и модели анализа данных: OLAP и Data Mining. – СПб.:Петербург, 2004 - 336 с.

2. *Чубукова И.А.* Data Mining: учеб. пособие – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 382 с.

3. *Сегаран Т.* Программируем коллективный разум. – СПб:Символ-Плюс, 2008. – 368 с.