УДК 004.42

И. В. Хмелева, Т.Г. Турчанова, А.Намазбек у, М. В. Коржов
hmelevai@gmail.com, turchan@mail.ru, n.u.abdysamat@gmail.com, megasdev@gmail.com
Кыргызско-Российский (Славянский) университет Бишкек, Кыргызстан

РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ КАК РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ МЕТОДА К-СРЕДНИХ

В статье описано разработанное мобильное приложение для навигации по культурным мероприятиям города Бишкек. Представленная работа является одним из возможных способов реализации рекомендательной системы на основе метода *k*-средних.

Ключевые слова: метод к-средних, рекомендательные системы, тэг, порог гравитации.

Введение. В нашу жизнь устойчиво вошли мобильные технологии, которые кардинальным способом улучшают как процессы производства, так и процессы потребления информации. Использование мобильных технологий позволяет быть в курсе всех событий в мире, прилагая для этого минимум усилий. Мобильные технологии позволяют снизить стоимость продукции для конечных потребителей за счет оптимизации процессов, сокращения производственных издержек и непроизводственных затрат. Кроме того, мобильные приложения пользуются большим успехом среди молодежи, в связи с чем увеличился спрос на их разработку. Мобильные приложения используют и как рекламные ролики, постоянно обновляемые, и как интернет-магазины, и просто как гаджеты или игровые программы. Среди коммерческих компаний такие приложения весьма актуальны и перспективны. В больших городах очень популяры приложения-навигаторы по мероприятиям, проводимым в городе, это удобно и для жителей города и для туристов. Подобные приложения разработаны для больших городов России, таких как Москва, Санкт-Петербург, Екатеринбург и других.

Требовалось разработать мобильное приложение для навигации по культурным мероприятиям города Бишкек. Приложение должно работать на любых моделях мобильных устройств.

1. Обзор алгоритмов поиска информации.

В глобальной сети сложно найти нужные данные. Для упрощения процедуры поиска разработаны различные поисковые алгоритмы и системы, которые на запрос пользователя выдают ссылки на страницы с информацией, указанной в запросе. Развитие социальных сетей усовершенствовало механизмы поиска до выработки рекомендаций [1] для конкретного пользователя на основе анализа его предыдущих запросов, а это предполагает применение интеллектуального анализа данных.

На сегодняшний день одним из вариантов разработки рекомендательных систем является использование методов коллаборативной фильтрации (КФ). Коллаборативная фильтрация – класс методов построения рекомендаций (прогнозов) на основе известных предпочтений (оценок) группы пользователей.

Основная идея алгоритмов КФ заключается в предложении новых элементов для конкретного пользователя на основе предыдущих его предпочтений или мнения других его единомышленников. К настоящему времени разработан целый ряд алгоритмов КФ [2-

5], которые можно разделить на следующие категории:

1. Методы, основанные на анализе имеющихся оценок, – анамнестические методы
    a) методы на основе сходства пользователей;
    b) методы на основе сходства элементов.
2. Методы, основанные на анализе модели данных, – модельные методы
    a) методы Байесовых сетей;
    b) методы кластерного анализа;
    c) методы на основе Марковских моделей;
    d) методы латентного семантического анализа;
    e) сингулярное разложение;
    f) анализ главных компонент.
3. Методы, основанные на объединении предыдущих алгоритмов, – гибридные методы.

Целью методов первой группы является объединение схожих объектов в группы на основе матрицы оценок [2-4]. Эти методы имеют высокую точность, но весьма ресурсоемки и имеют ограниченные возможности при обработке больших объемов данных. Кроме того, хранение всей матрицы предпочтений во многих случаях избыточно: например, многие фантастические фильмы будут нравиться определенной группе пользователей в равной степени. Поэтому возникает задача понижения размерности матрицы оценок. Такие задачи решают методы второй группы.

В этом случае возможен вариант объединения пользователей в кластеры с помощью некоторого индекса сходства. Элементы и оценки, данные пользователями из одного кластера, используются для вычисления рекомендаций. Кластерные модели лучше масштабируются, т.к. сверяют кластер пользователя с относительно небольшим количеством сегментов, а не с целой пользовательской базой.

В поставленной задаче использовался метод *k*-средних для формирования рекомендаций пользователям мобильного приложения – навигатора.

2. Разработка диаграммы развертывания приложения.

Программное приложение предназначено для навигации по культурным мероприятиям города. На рисунке 1 приведена диаграмма развертывания приложения, из которой видно, что приложение имеет двухуровневую архитектуру, обработка данных ведется на сервере, чтобы не перегружать «клиента» и пользоваться приложением с мобильных устройств старой версии.

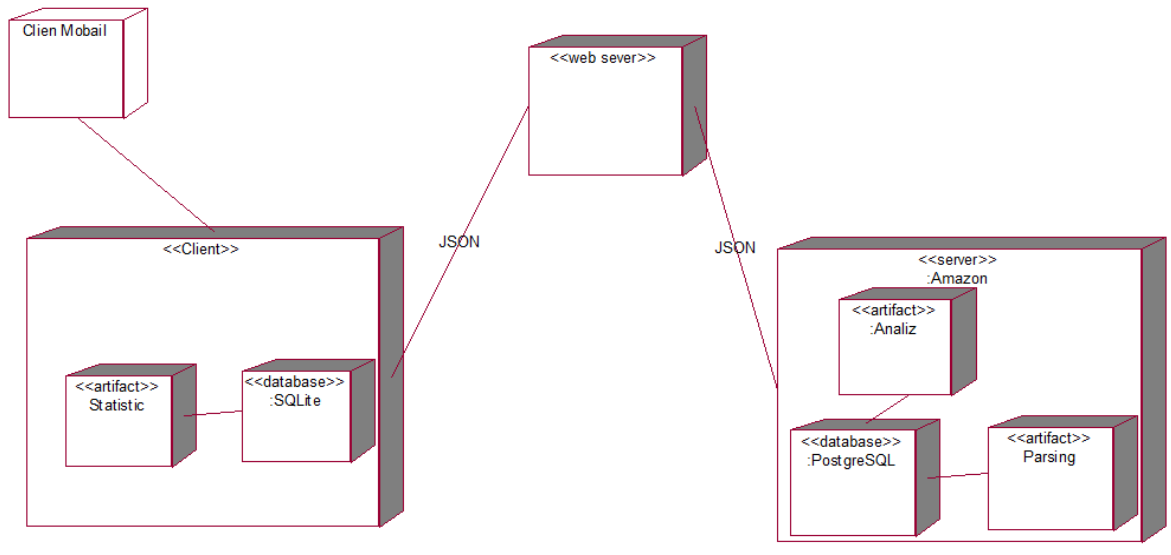

Рис.1. Диаграмма развертывания.

Рассмотрим подробнее задачи каждой стороны навигатора.

3. Описание работы клиентской части приложения.

Перед клиентским приложением стоит задача сбора и анализа данных с различных устройств пользователей и передача этих данных на сервер для дальнейшего анализа и выработки рекомендаций. Процесс сбора анонимной статистики выполняется пакетом средств разработчика (SDK) в фоновом режиме и не затормаживает работу приложения.

Артефакт Parsing реализует автоматический сбор информации обо всех мероприятиях города с сайтов организаций (аналог RSS) и собирает статистику о посещаемости и предпочтениях пользователей приложения. Данные передаются в формате JSON [6], что удобно и для передачи, и для дальнейшей обработки.

Клиентское приложение раз в сутки получает готовую информацию с сервера. Алгоритм сбора статистики о посещаемости и предпочтениях пользователей включает в себя несколько шагов:

1) для каждого посетителя генерируется персональный номер, к которому прикрепляются идентификаторы мероприятий и частота посещения каждого;

2) для учёта посещаемости к каждому пользователю также крепятся его страна, модель устройства, размер экрана, состояние сети, оперативная память и множество других параметров (Рис.2);

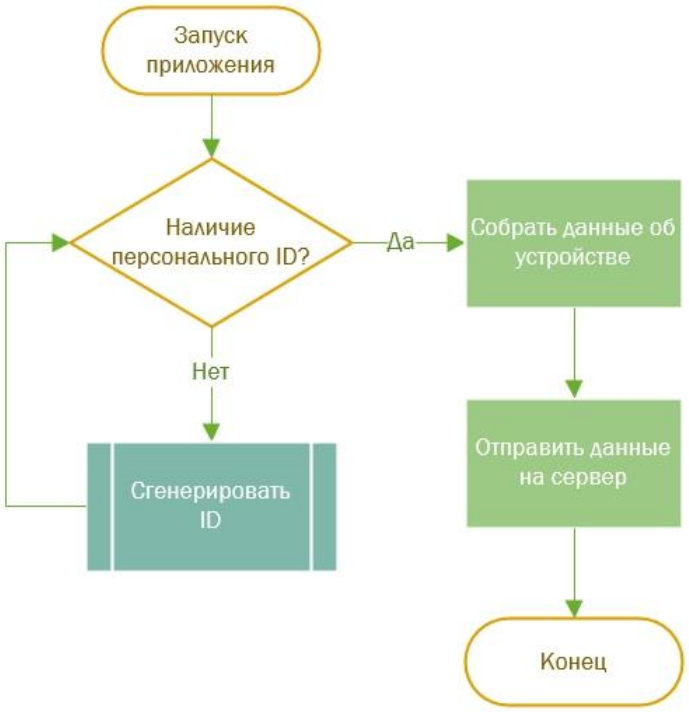

Рис.2. Алгоритм сбора информации об устройстве пользователя.

3) подготовленные на предыдущих шагах данные выгружаются на сервер, где производится их обработка (выделение тематики события, степени заинтересованности и др.) и учет при последующей отдаче контента пользователю. Стоит учесть, что на этом этапе в журнал устройства записываются практически все действия, которые пользователь произвёл в

приложении: эта информация отправляется на сервер вместе со служебной из п.2 (Рис.3)

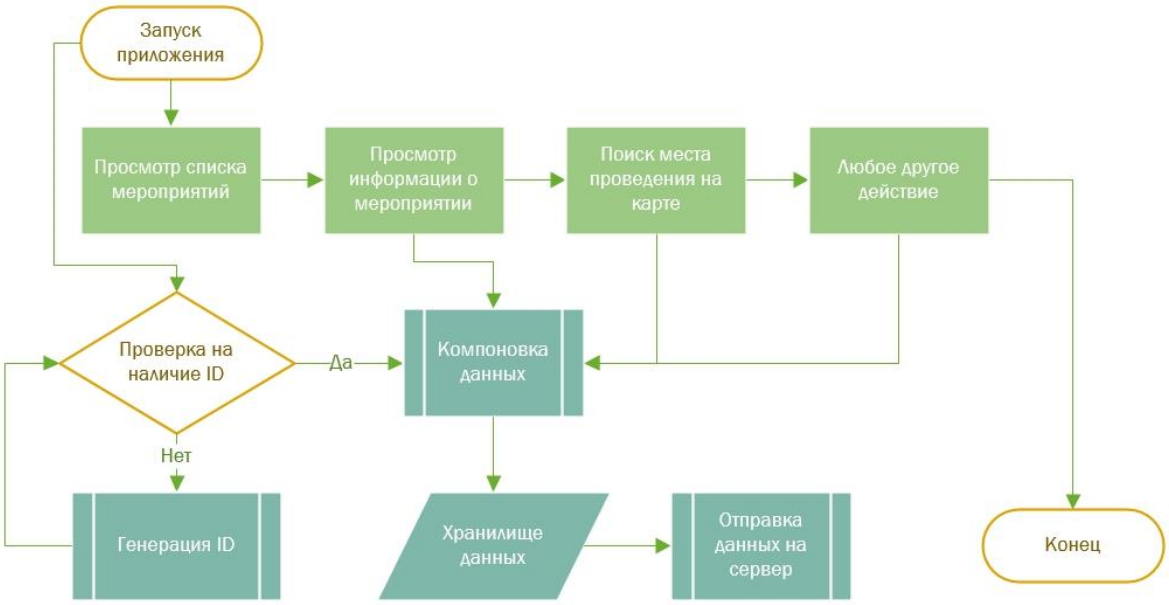

Рис.3. Алгоритм сбора информации о действиях пользователей в приложении.

Функция рассылки персональных уведомлений позволяет пользователю получать уведомления о мероприятиях, которые могут его заинтересовать, и напоминает ему о предстоящем событии. После активации функции сервер ставит пользователя на учёт и сообщает ему всю необходимую информацию. Алгоритм приёма и показа уведомления продемонстрирован на Рис.4:

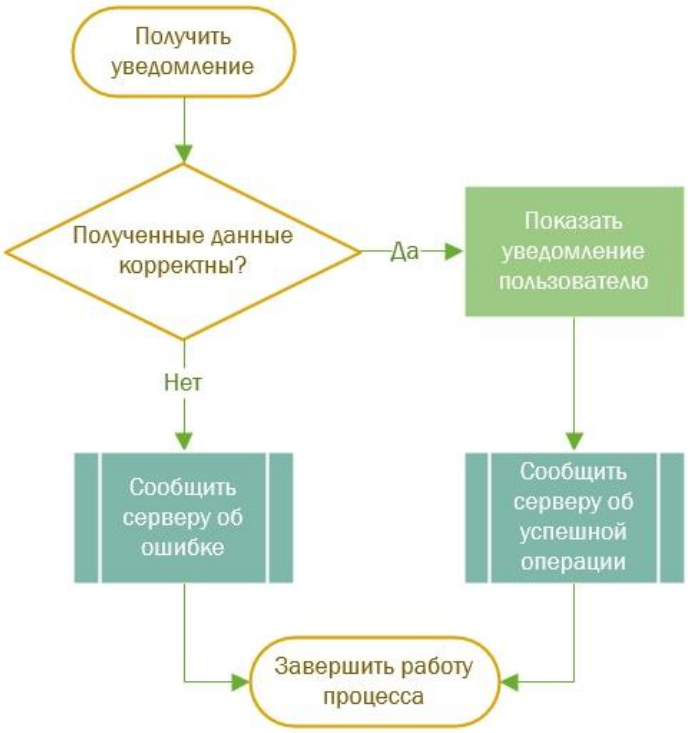

Рис.4. Процесс получения уведомления о мероприятии с сервера.

Основное требование при разработке мобильных систем заключается в том, что мобильное устройство не должно выполнять никаких вычислительных действий - это задача сервера. Устройства могут быть старыми, в которых приложение может "упасть" даже при минимальных нагрузках. Соответственно, в задачи клиентской части приложения входит только получение данных, их демонстрация, сбор базовой информации об устройстве и пользователе и отправка данных на сервер.

4. Описание работы серверной части приложения.

Сервер помимо хранения данных выполняет их анализ для выработки рекомендаций каждому пользователю. Интеллектуальный анализ данных решает задачу кластеризации, которая заключается в делении множества объектов на группы (кластеры) схожих по параметрам. При этом, в отличие от классификации, число кластеров и их характеристики могут быть заранее неизвестны и определяться в ходе построения кластеров исходя из степени близости объединяемых объектов по совокупности параметров[7]. Для решения задачи кластеризации применяется комбинированный метод k-средних.

4.1. Применение алгоритма k-средних для решения поставленной задачи.

Алгоритм k-средних является простым повторяющимся алгоритмом кластеризации, который разделяет определенный набор данных на заданное пользователем число кластеров k. Алгоритм прост для реализации и запуска (вычислительная сложность алгоритма: $O(nkl)$, где k – число кластеров, ℓ – число итераций), относительно быстрый, легко адаптируется и распространен на практике [7].

Рассмотрим алгоритм k-средних применительно к поставленной задаче.

Пусть X есть множество всех наблюдаемых объектов $x_i \in X, 1 \le i \le nof(X)$, отнесенных к одному из кластеров $X_l \subseteq X, 1 \le l \le nof(K)$, где $K = \{X_1, \dots, X_l, \dots, X_{nof(K)}\}$ – множество всех сформированных кластеров. Объекты обладают набором определенных характеристик (страна, модель устройства, размер экрана, состояние сети, оперативная память, тема события, количество запросов и множество других параметров). Характеристику объекта назовем тегом (обозначим T_i). Тогда пользователя можно представить в виде n-мерного вектора тегов: $U = [T_1, T_2, \dots . T_n]$, где T это есть тег, который описан двумерным вектором: $T_i = [F, R]$, где i-идентификатор тега, F – частота запросов тега пользователем, R – количество обнулений, $n \le N$, N – общее количество тегов в базе данных.

Тег имеет численный показатель притяжения G- «сила гравитации». Тег – также является центром притяжения других тегов, в данном случае спутников. Центр тяжести можно представить N-мерным вектором тегов: $G = [T_1, T_2, \dots T_n]$, где $n \le N$, N – общее количество тегов в базе данных. В качестве меры близости использовано Евклидово расстояние[7].

Для определения схожести тегов между собой введены понятия «достоверных данных» и «недостоверных данных».

Достоверные данные – это данные, которые принимаются как заведомо истинные. В данном случае это данные, которые принимаются сервером при создании события пользователем. Создание события всегда сопровождаются наличием тегов, и если количество тегов при создании события больше одного, то теги связываются между

собой, то есть, относительная «сила гравитации» между ними возрастает на единицу.

Недостоверные данные – это данные, которые не проходят алгоритм проверки на определение схожести. В данном случае к недостоверным данным можно отнести запросы пользователя. Пользователь может вводить разные запросы в поиске, и нельзя достоверно сказать о том, связан ли текущий запрос с последующим или с предыдущим. Для этого вводится понятие «Порог гравитации» – численный показатель частоты набора пользователем конкретного тега.

4.2. Результаты работы алгоритма k-средних.

Пример визуализации работы алгоритма для объекта «театр» приведен на рис. 5.

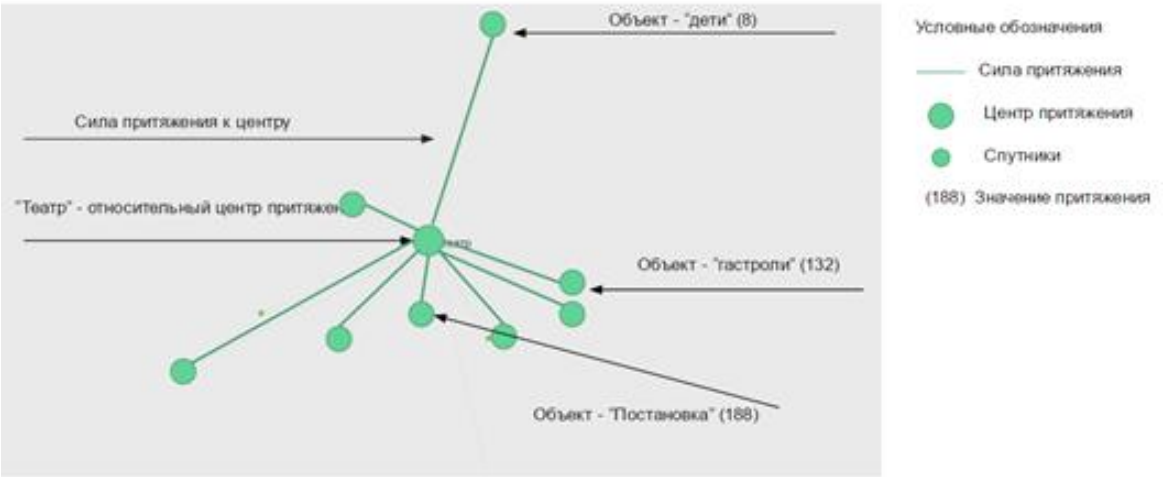

Рис. 5. Визуализация объекта «Театр».

Здесь пользователь представлен в виде вектора тегов $U = [T_1, T_2, T_3]$, Порог гравитации равен 10, $T_1 = [10,0]$, $T_2 = [2,0]$, $T_3 = [8,0]$. Следует обратить внимание на первые аргументы векторов: (10,2,8). Когда порог гравитации достигает границы, что четко наблюдается у $T_1 = [10,0]$, то данный тег обнуляется, связавшись с ближайшим тегом $T_3 = [8,0]$.

После проведения данной операции теги пользователя будут иметь вид: $T_1 = [0,1]$, $T_2 = [2,0]$, $T_3 = [8,0]$, причем тег T_3 не обнулился.

На рисунке 5 стоит обратить внимание на объект «дети», который связан с объектом «театр» крайне слабо – 8 единиц притяжения. Такой результат можно было бы отнести к погрешности, однако, в ходе анализа данных было выявлено, что в театрах проходили детские утренники и спектакли, так что объекты «дети» и «театр» связаны закономерно.

Таким образом, разработано мобильное приложение для поиска культурных мероприятий города Бишкек с учетом предпочтений пользователя. Механизм формирования предпочтений основан на реализации кластерного анализа методом *k*-средних. Разработанный сервис запущен в пользование в августе 2016 г с рабочим названием «CitySpy».

Литература

1. М. Тим Джонс, Рекомендательные системы. //URL: https://www.ibm.com/developerworks/ru/library/os-recommender1/ (дата обращения 25.03.2017).
2. J.S. Breese, D. Heckerman, and C. Kadie, –Empirical Analysis of Predictive Algorithms for Collaborative Filtering// Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998.
3. Xiaoyuan Su and Taghi M. Khoshgoftaar "A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques" // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA : 2009. – С. 1–19.
4. G. Adomavicius На пути к новому поколению рекомендационных систем: обзор имеющихся систем и возможные инновации.// IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, Июнь 2005 URL: http://artpragmatica.ru/rs/in/pic/58-870-20061024072441-Toward_the_next_genera-tion_ of_recommender_systems.doc
5. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов // Труды ИСП РАН. 2012, Т.22, с. 401–418 URL: http://cyberleninka.ru/article/n/sistemy-rekomendatsiy-obzor-sovremennyh-odhodov (дата обращения 20.03.2017).
6. Язык JavaScript. Электронный учебник. // URL: https://learn.javascript.ru/json (дата обращения 19.04.2017).
7. Портал знаний. Кластеризация: метод к-средних. //URL: http://statistica.ru/theory/klasterizatsiya-metod-k-srednikh/ (дата обращения 21.03.2017).