Санкт-Петербургский государственный университет
Математико-механический факультет

Кафедра Системного Программирования

Лобанов Артём Алексеевич

Применение неевклидовых методов кластеризации для определения популярных маршрутов абонентов

Курсовая работа

Научный руководитель:
ведущий разработчик ООО «НМТ» Невоструев К. Н.

Санкт-Петербург
2016

Оглавление

Введение

В последнее время сильно возросла конкуренция на рынке телекоммуникационных услуг. Компании-поставщики услуг прилагают максимум усилий, чтобы заполучить больший процент рынка. С развитием систем хранения и обработки информации компании получили доступ к огромному количеству ресурсов, которые могут быть использованы для повышения конкурентноспособности. В то же время сейчас наблюдаются значительные прорывы в области интеллектуальной обработки данных. Как следствие, вопросы внедрения в производство научных методов анализа информации становятся все более актуальными.

Среди важнейших задач, обеспечивающих повышение дохода компании на рынке, можно выделить задачу привлечения новых клиентов: необходимо находить способы, посредством которых потенциальные клиенты смогут узнать как можно больше информации о компании. Один из методов решения данной задачи – анализ информации о перемещениях пользователей. Данные, получаемые в результате такого анализа, могут использоваться для определения выгодных расположений рекламы компании и многих других целей.

Из большого количества специализированных методов для решения задачи определения структуры пользовательских маршрутов наиболее подходящим является кластерный анализ: маршруты клиентов не обладают заранее установленными метками, поэтому методы машинного обучения с учителем (supervised learning) в данной области менее применимы.

В данной работе представлен метод определения распространенных маршрутов абонентов крупного российского мобильного оператора. Основная его часть заключается в подготовке данных для кластеризации и использовании неевклидовых методов кластеризации в пространстве, имеющих ряд преимуществ в подобных задачах с неопределённой размерностью исходных данных.

1. Предметная область и постановка задачи

1.1. Общая терминология

Задача *кластеризации* (*кластерного анализа*) заключается в следующем. Пусть X – множество объектов, Y – множество меток кластеров. Задана функция расстояния между объектами $\rho(x_i, x_j)$, а также может быть задача обучающая выборка $X^m = \{x_1, ..., x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества C_i, называемые *кластерами*, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров значительно отличались. При этом каждому объекту $x_i \in X^m$ присваивается метка кластера y_i.

Алгоритм кластеризации – функция $a : X \to Y$, которая любому объекту $x \in X$ ставит в соответствие метку кластера $y \in Y$. Множество меток Y при различных постановках задачи может быть известно заранее, а может определяться в процессе работы алгоритма с точки зрения того или иного критерия качества кластеризации.

Кластеризация является частным случаем обучения без учителя (unsupervised learning) и отличается от классификации тем, что метки исходных объектов y_i изначально не заданы, и даже само множество Y может быть неизвестно.

1.2. Кластеризация в неевклидовом пространстве

Выделяют методы кластеризации, которые не требуют евклидовой метрики для определения близости объектов. В случае применения таких методов, как правило, для всех объектов x_i используется заранее заданная матрица схожести (различия) $M : M_{ij} = \rho(x_i, x_j)$; кроме того иногда используются статистические критерии определения схожести, например, для некоторых алгоритмов достаточно информации о том, какова вероятность того, что данный объект x_i принадлежит кластеру y_i: $p_{ij} = P\{x_i \in C_i\}$ [1].

Неевклидовы методы кластеризации часто применяются в случаях, когда исходные данные имеют различную размерность, поэтому не могут интерпретироваться, как точки $(x_1, ..., x_n) \in X^n$ какого-либо евклидового пространства X^n. Стоит заметить, что практически любую задачу кластеризации можно преобразовать к задаче, решаемой евклидовыми методами, однако это повлечет за собой неизбежные потери качества входных данных, в то время как неевклидовы методы позволяют проводить кластеризацию на изначальных данных.

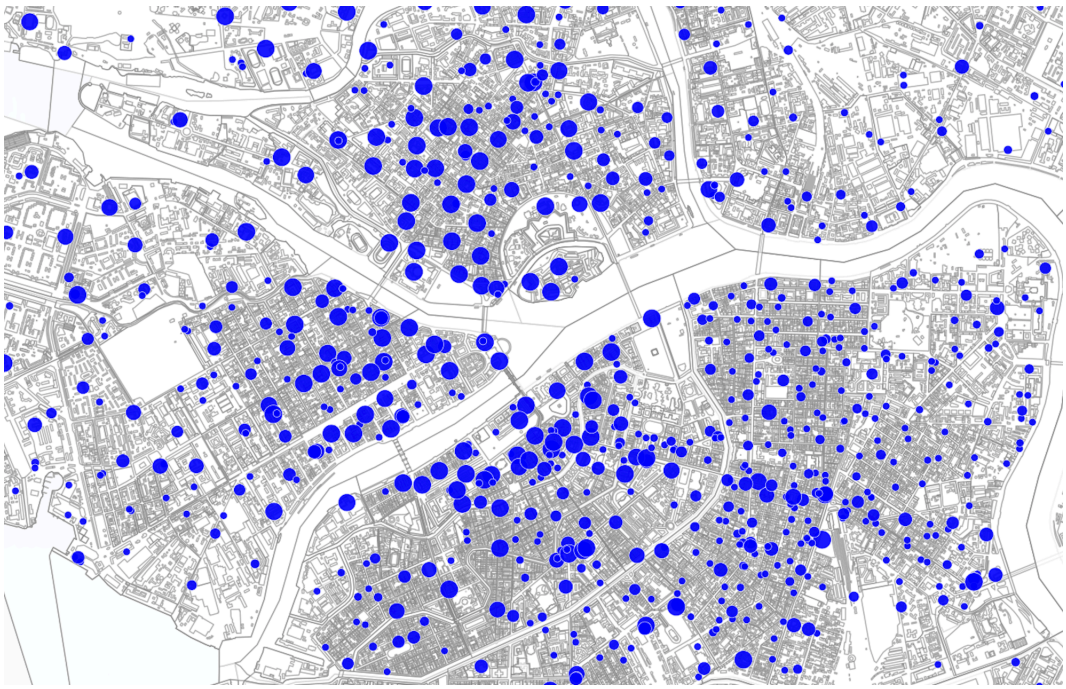

Рис. 1: Данные о базовых станциях

1.3. Исходные данные

Исходные данные предоставлены одной из крупнейших российских телекоммуникационных компаний. Данные содержат информацию о базовых станциях (base stations) – приёмопередатчиках радиосигнала, относительно которых определяется местоположение абонентов, а также информацию о пользовательской активности в различные дни спортивных событий с октября по декабрь 2015 года.

Для каждой базовой станции известны ее идентификатор (комбинация LAC – Local Area Code и Cell ID), координаты и адрес.

Для каждой единицы пользовательской активности предоставлены:

- идентификатор пользователя (user ID);
- тип активности (входящие/исходящие звонки/SMS, пакетная передача данных);
- время активности;
- идентификатор базовой станции, к которой пользователь находился ближе всего во время активности.

Данные содержат информацию о 3230 базовых станций, 24 000 пользователей и более, чем 10 000 000 единиц пользовательской активности.

1.4. Постановка задачи

Целью данной работы является разработка алгоритма определения распространенных маршрутов абонентов.

Задачи, решаемые в рамках работы:

- Обработать данные о пользовательской активности
- Провести кластеризацию маршрутов
- Определить приоритетные места расположения рекламы

2. Обработка данных

2.1. Выделение маршрутов

Пусть S – множество базовых станций, C – координатное пространство. Набор исходных данных представляет собой набор *activities* векторов

$$\langle userid,\ activity\ type,\ time,\ base\ station\ id \rangle$$

Кроме того известно соответствие базовых станций их координатам:

$$coords : S \to C$$

Для построения маршрутов пользователя за какой-либо промежуток времени рассматривались координаты всех активностей пользователя по возрастанию времени активности:

$$route(userid) = \{(lat_1, lon_1), ..., (lat_n, lon_n)\} \Leftrightarrow$$
$$\forall i\ \exists\ a_i \in activities\ \ coords(a_i) = (lat_i, lon_i)$$

При такой генерации маршрутов для каждого спортивного события генерируется примерно 7000 маршрутов.

При построении было удалено большое количество повторяющихся данных о базовых станциях, в связи с чем реальное их количество было уменьшено с 49000 до 3230 станций без какой-либо потери информации.

2.2. Визуализация маршрутов

В ходе работы было принято решение реализовать визуализацию данных о путях для более качественного анализа. Визуализация реализована двумя способами:

- Визуализация непосредственно маршрутов
- Визуализация путей по улицам, соответствующих данным маршрутам

Визуализация непосредственно маршрутов была реализована с помощью библиотеки Basemap (расширение Python-библиотеки matplotlib); в качестве географических данных были использованы *шейп-файлы* (shapefiles), основанные на базе данных OpenStreetMap [2]. Такая реализация не требует постоянного доступа к серверу карт, поэтому была использована на протяжении большинства экспериментов.

Визуализация путей по улицам была реализована с помощью Google Directions API и библиотеки Python Client for Google Maps Services. При вызове этой библиотеки данный маршрут уточняется, полученный маршрут интерпретируется как реальный

маршрут абонента по улицам. Из-за описанных свойств данный вид визуализации используется на последнем этапе для определения потенциальных мест размещения рекламы.

2.3. Подготовка данных о маршрутах для кластеризации

Ввиду того, что кластеризация должна проводиться на маршрутах к/от стадиона, необходимо было преобразовать полученную группу маршрутов для получения маршрутов к/от стадиона.

Было выявлено несколько различных преобразований множества маршрутов $R = \{route_1, ..., route_n\}$, позволяющих существенно улучшить качество маршрутов:

1. **Фильтрация**

- по посещению маршрутом стадиона:

$$\exists\, coord_j \in route_i \mid coord_j = coord_{stad}$$

- по наибольшей длине отрезка маршрута:

$$max_{0 \le j \le |route_i|-1} dist(coord_j, coord_{j+1}) < \lambda_1$$

  Данный вид фильтарции обусловлен тем, что слишком большое расстояние между точками маршрута является признаком недостаточного количества исходных данных для маршрута или их некоректности.

- по углу между радиус-векторами маршрута (радиус вектор маршрута $rv_i(route)$ – вектор вида $route_i - coords_{stad}$ такой, что $route_i \in route$ и $route_i \neq coords_{stad}$):

$$max_{0 \le j,k \le |route_i|} \angle(rv_j, rv_k) < \lambda_2$$

  Данное преобразование набора маршрутов уместно применять после остальных с целью потери меньшего количества данных. Оно мотивировано наблюдением, что маршруты со слишком большим разбросом направлений отрезков практически не поддаются кластеризации и дают мало информации о потенциальных местах для размещения рекламы.

2. **Разбиение маршрутов на маршруты к/от стадиона**

- "Раскрытие" длинных циклов вокруг стадиона

  Если в маршруте существует существует цикл – подмаршрут вида $\{coord_{stad}, ..., coord_{stad}\}$ длины больше данного параметра λ_3, то находится наиболее дальняя точка цикла

$$coord_j \mid \forall\, k\ dist(coord_{stad}, coord_k) \leqslant dist(coord_{stad}, coord_j),$$

и затем на основе данного цикла строятся два новых маршрута $\{coord_{stad}, ..., coord_j\}$ и $\{coord_j, ..., coord_{stad}\}$.

- Исключение остальных циклов

  При наличинии в маршруте циклов длины меньше λ_3, такие циклы просто игнорируются, то есть на основе каждого маршрута с циклом $\{coord_1, ..., coord_j, coord_{stad}, ..., coord_{stad}, coord_k, ..., coord_n\}$
  строится маршрут $\{coord_1, ..., coord_j, coord_{stad}, coord_k, ..., coord_n\}$.

При таком преобразовании, проведенном на выборке из 21 000 маршрутов, было обнаружено, что группы маршрутов к стадионам и от стадионов примерно одинаково распределены, ввиду чего было решено рассматривать все маршруты, как маршруты от стадионов.

2.4. Результаты обработки данных о маршрутах

После анализа данных о маршрутах была получена гораздо более качественная выборка (см. Рис. 2). В ходе работы использовались параметры $\lambda_1 = 3$ мили, $\lambda_2 = 1.2$ радианов, $\lambda_3 = 4$, при которых количество полученных после обработки маршрутов равнялось примерно 40% от изначальной выборки; данный процент можно значительно увеличить, в первую очередь, за счет варьирования параметра λ_1, однако во время экспериментов использовалось небольшое значение данного показателя для лучших результатов кластеризации.

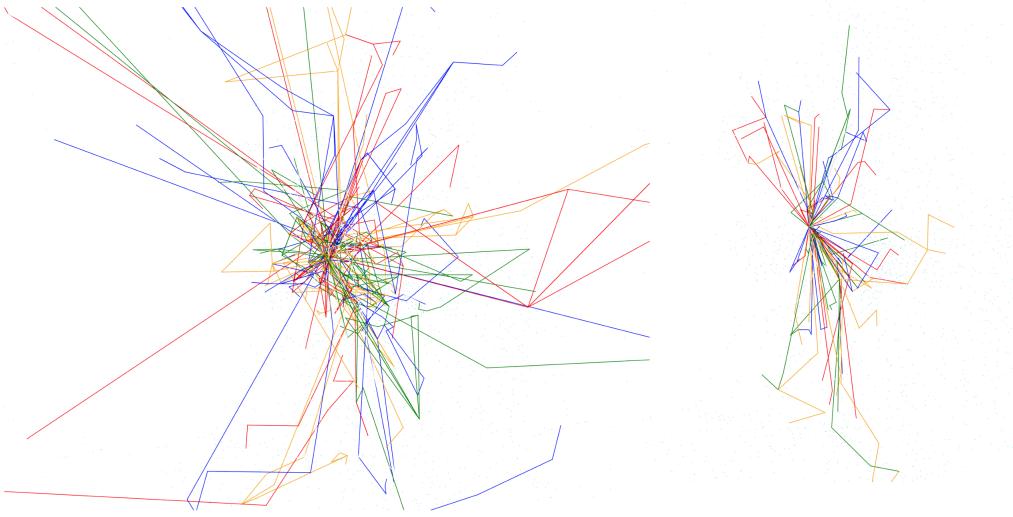

Рис. 2: Силуэты 100 случайных маршрутов до и после обработки данных

3. Кластериация

Практически все широко используемые алгоритмы кластеризации в пространстве подразделяются на 2 класса:

1. **Иерархические**. Алгоритмы, основанные на построении по выборке древовидных структур, отражающих сходство объектов. По получившейся структуре определяются итоговые кластеры.

2. **Point Assignment**. Алгоритмы, предполагающие изначальное присвоение объектов k кластерам. Такие алгоритмы пошагово выполняющют пересчет внутренних метрик и последующее переприсвоение объектов кластерам до достижения заданного порога изменения метрик.

В данной работе рассмотрены 3 метода кластеризации:

- Агломеративная кластеризация с неевклидовыми метриками
- GRGPF
- EM-алгоритм

Первые 2 метода относятся к иерархическим, последний – к point assignment методам.

Для применения к данным а маршрутах абонентах в случае агломеративной кластеризации была использована реализация из библиотеки scikit-learn, в случае EM-алгоритма – из библиотеки OpenCV, алгоритм GRGPF же ввиду отстутствия полноценных применимых к данной задаче реализаций был реализован самостоятельно.

Далее в данном разделе рассматриваются различные метрики близости маршрутов, а также описываются общие принципы работы вышеприведённых методов, исключение – EM-алгоритм, использовавшийся без специализированных метрик [3].

3.1. Метрика близости маршрутов

Для задачи кластеризации маршрутов была выявлена эффективная мера близости маршрутов $sim(route_i, route_j)$: пусть задана последовательность граничных значений $\mu_1 \leqslant \mu_2 \leqslant ... \leqslant \mu_n$; пусть, кроме того, задана характеристическая функция на отрезке

$$\mathbb{1}_{[a,b]}(x) = \begin{cases} 1, & a \leqslant x \leqslant b \\ 0, & otherwise \end{cases}$$

Пусть для данного маршрура $route = \{x_n, ..., x_n\}$ $d_{ij} = dist(x_i, x_j)$, задана близость отрезков маршрута $sd(x_1, x_2, y_1, y_2) \stackrel{\text{def}}{=} sd(x_1, y_1) = \min(\max(d_{11}, d_{22}), \max(d_{12}, d_{21}))$ и среднее расстояние от точек до стадиона $dist_{stad}(i, j) = 0.5 \cdot (dist(x_i, coord_{stad}) + dist(x_j, coord_{stad}))$.

Тогда функция близости маршрутов:

$$sim(route_1, route_2) = \sum_{\substack{x_i \in route_1 \\ x_j \in route_2}} dist_{stad}(i,j) \sum_{1 \leqslant i \leqslant n-1} \omega_i \mathbb{1}_{[\mu_i, \mu_{i+1}]}(sd(x_i, x_j))$$

При этом веса ω_i выбираются эвристически.

Также были рассмотрены стандартные метрики близости маршрутов, такие, например, как мера Жаккара, однако они содержат слишком мало информации о строении путей, поэтому гораздо менее применимы для данной задачи [4].

3.2. Метрики близости кластеров

Пусть $C = \{c_1, ..., c_n\}$, $D = \{d_1, ..., d_m\}$ – кластеры.

$rowsum(c_i)$ – по определению сумма квадратов расстояний от данной точки c_i до остальных точек C:

$$rowsum(c_i) = \sum_{c_j \in C} \rho^2(c_i, c_j).$$

Кластроид – точка кластера, минимизирующая $rowsum$:

$$\widetilde{C} = c_i \in C \mid \forall\, c_j \in C \; rowsum(c_j) \geqslant rowsum(c_i).$$

В большинстве алгоритмов кластеризации в пространстве существует необходимость каким-то образом вычислять расстояние между кластерами. В случае неевклидовых методов одним из самых распространенных методов является метод, основанный на кластроидах – расстояние между кластерами определяется расстоянием между их кластроидами:

$$\rho(C, D) \stackrel{\text{def}}{=} \rho(\widetilde{C}, \widetilde{D}). \tag{1}$$

3.3. Агломеративная кластеризация

При агломеративной кластеризации сначала имеется n объектов $X_1, ..., X_n$ и n содержащих их одноэлементных кластеров $C_1, ..., C_n$.

На каждом шаге алгоритма считается метрика $\rho(C_i, C_j)$ расстояния между кластерами; в случае неевклидовой кластеризации, основанной на кластроидах, используется функция (1).

После этого находятся 2 самых близких по данной метрике кластера C_i, C_j и объединяются в один:

$$C_{new} = \bigcup \arg\min_{\{C_i, C_j\}} \rho(C_i, C_j)$$

Процесс продолжается до тех пор, когда в текущем наборе кластеров останется только 1 кластер, содержащий все объекты.

3.4. GRGPF

GRGPF – один из алгоритмов, разработанных специально для кластеризации в многомерном неевклидовом пространстве [5]. Он также адаптирован для очень больших объёмов данных, т. к. предполагает, что не все данные могут поместиться в оперативной памяти компьютера.

Алгоритм поддерживает структуру данных, аналагичную R-дереву, которая устроена следующим образом: в листьях хранятся признаки (features) кластеров, которые задают общее их описание. Используется заданный набор признаков:

- количество точек в кластере, N;
- кластроид данного кластера, $\widetilde{C}$;
- p точек, наиболее близких к кластроиду;
- p точек, наиболее далёких от кластроида.

Во внутренних узлях хранятся образцы кластроидов кластеров, представленных наследниками узла.

Добавление точек в GRGPF. По ходу работы алгоритма в дерево добавляются новые точки p: происходит спуск по дереву, данная точка добавляется в кластер C_p с ближайшим к p кластроидом:

$$C_p = \arg\max_C \rho(\widetilde{C}, p).$$

После добавления оценивается $rowsum(p)$:

$$rowsum(p) = rowsum(\widetilde{C}) + N\, d^2(p, \widetilde{C}) \tag{2}$$

Такая оценка обусловлена т. н. «проклятьем размерности» – известным наблюдением о том, что в многомерных пространствах все векторы практически ортогональны [6]. В данном предположении формула (2) легко обосновывается теоремой Пифагора.

При необходимости дерево балансируется по алгоритму, аналогичному балансировке R-дерева.

Разбиение и слияние кластеров в GRGPF.

Алгоритмом GRGPF задается порог радиуса кластера, по достижению которого кластер разбивается на 2 части: для этого весь кластер загружается в основную память, разделяется с условием минимизации $rowsum$, признаки заново считаются для двух получившихся кластеров;

Похожие кластеры по ходу работы алгоритма похожие кластеры C_i и C_j объединяются в кластер C. Для определения кластроида нового кластера оценивается $rowsum$

для всех $X \in C_i \cup C_j$: Пусть x, точка кластера C_i:

$$rowsum(x) = rowsum_i(x) + N_j(D^2(x, \widetilde{C}_i) + D^2(C_i, C_j)) + rowsum_j(\widetilde{C_j})$$

Эта оценка также обусловлена «проклятьем размерности»: первое слагаемое отвечает за расстояния от x до остальных $x' \in C_i$, второе – за расстояние между $\widetilde{C}_i$ и $\widetilde{C_j}$, третье – за расстояния от $\widetilde{C_j}$ до всех $y \in C_j$.

3.5. Оценка качества кластеризации

Полученные в результате кластеризации кластеры могут быть оценены визуально: маршруты в одном кластере схожи по своему строению и направлению (см. Рис 2), на основании этих данных можно делать содержательные выводы о потенциальных местах расположения рекламы.

Однако результирующие кластеры можно оценить качественно: несмотря на то, что решение задачи кластеризации принципиально неоднозначно, существуют различные характеристики, по которым можно определить качество полученных кластеров. Одними из классических и в то же время наиболее часто используемых примеров таких характеристик являются квадраты расстояний между объектами кластера (SSQ), радиус и диаметр кластера [7]:

$$SSQ(C) = \sum_{x,y \in C} \rho^2(x,y),\ r(C) = \sqrt{\frac{\sum_{x \in C} \rho^2(\widetilde{C}, x)}{N}},\ d(C) = \sqrt{\frac{\sum_{x \in C} \sum_{y \in C} \rho^2(x,y)}{N(N-1)}}.$$

Ниже на Рис. 3 представлены графики SSQ и радиусов кластеров для различных алгоритмов на нескольких выборках.

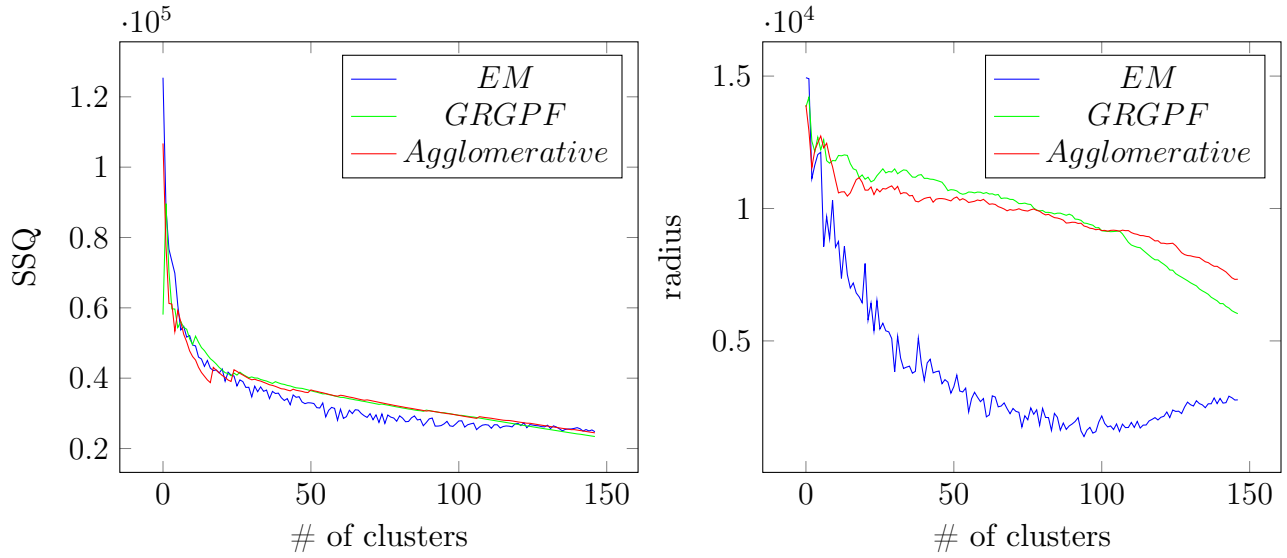

Рис. 3: rowsum и радиусы кластеров в зависимости от их количества

Можно отметить, что EM гораздо менее стабилен при данной метрике, чем 2 других алгоритма, GRGPF же наиболее устойчив к изменению строения кластеров. Однако все 3 алгоритма позволяют установить примерное количество реальных кластеров на данной выборке – 25-30 кластеров.

Наложение маршрутов на карту.

Было проведено наложение результатов кластеризации на городские улицы для того, чтобы понять приоритетные направления абонентов. На Рис. 4 представлена карта, отражающая наиболее популярные маршруты абонентов, построенная на основе нескольких самых объемных кластеров. Кластеризация проводилась на выборке из 2400 маршрутов.

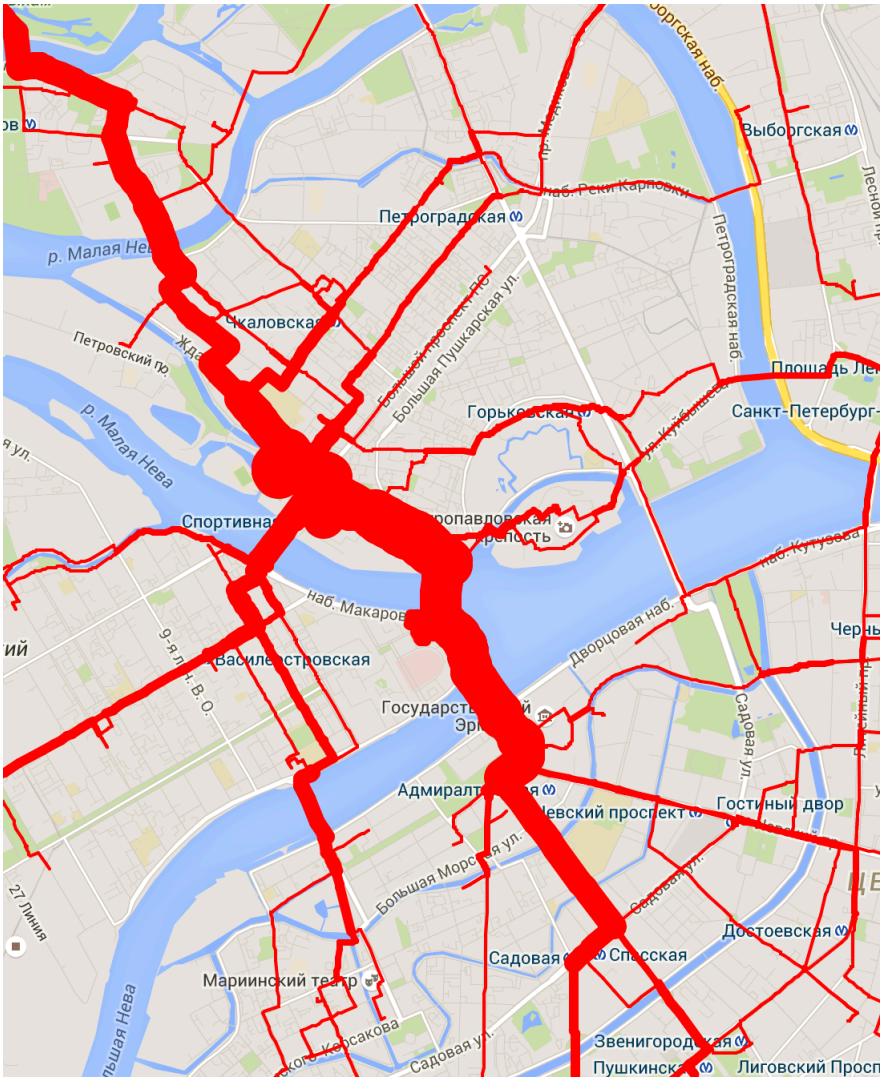

Рис. 4: Популярные маршруты на основе кластеризации

Маршруты, покрываемые объёмными кластерами.

Были проведены измерения эффективности кластеризации с точки зрения последующих затрат компании на рекламу. Результаты измерений показали, что 5 самых объёмных (содержащих наибольшее число маршрутов) кластеров из 27 покрывают почти треть маршрутов пользователей, а 10 самых объёмных – больше половины (см. Таблица 1). Соответственно можно использовать небольшое количество маршрутов, соответствующих объёмным кластерам, для информирования относительно значительной части аудитории.

# top clusters	03.10	20.10	24.10	31.10	21.11	24.11
top 5	0.32	0.35	0.32	0.34	0.27	0.31
top 10	0.57	0.50	0.52	0.55	0.52	0.56
top 15	0.73	0.64	0.69	0.71	0.75	0.75
top 27	1.0	1.0	1.0	1.0	1.0	1.0

Таблица 1: Процент маршрутов, покрываемых объёмными кластерами

Заключение

В ходе данной работы был разработан алгоритм определения распространённых маршрутов пользователей. Были решены следующие задачи:

- Проведены обработка исходных данных о пользователях, построение маршрутов, преобразование маршрутов в соответствии с поставленной задачей
- Выполнена кластеризация данных о маршрутах несколькими методами кластеризации в неевклидовом пространстве
- Реализован алгоритм GRGPF
- Проанализировано качество кластеризации и построена статистика, определяющая потенциальные места размещения рекламы

Список литературы

[1] Jundi Ding Ping Wang, Runing Ma. On non-euclidean metrics based clustering. *Springer Berlin Heidelberg*, 1999.

[2] Metro extracts – parts of the openstreetmap database for major world cities and their surrounding areas, 2013.

[3] C. Kotropoulos D. Ververidis. Gaussian mixture modeling by exploiting the mahalanobis distance. *IEEE Transactions on Signal Processing*, 2008.

[4] Holmes Finch. Comparison of distance measures in cluster analysis with dichotomous data. *Journal of Data Science 3*, pages 85–100, 2005.

[5] J.D. Ullman J. Leskovec, A. Rajaraman. *Mining of Massive Datasets.* Cambridge University Press, 2011.

[6] Vipin Kumar Michael Steinbach, Levent Ertöz. The challenges of clustering high dimensional data. pages 11–19.

[7] Jean-Charles Lamirel. Reliable clustering quality estimation from low to high dimensional data. *Springer International Publishing*, 2016.