

АНАЛИЗ ТРАЕКТОРИЙ В МОБИЛЬНЫХ СЕТЯХ НА БАЗЕ ИНФОРМАЦИИ О СЕТЕВОЙ БЛИЗОСТИ

Д. НАМИОТ, кандидат физико-математических наук
Факультет Вычислительной Математики и Кибернетики
МГУ им. М.В. Ломоносова
Ленинские горы, 119991 Москва, Россия
E-mail: dnamiot@gmail.com

М. ШНЕПС-ШНЕППЕ, доктор технических наук, профессор
Институт математики и информатики Латвийского университета
бульв. Райня 29, LV-1459 Рига, Латвия
E-mail: manfreds.sneps@gmail.com

Тема статьи относится к услугам мобильной сети, связанным с местоположением. Рассматривается задача выделения информации о перемещениях (траектории) из логов, относящихся к сетевой близости. Обычно, подобного рода извлечение (поиск) шаблонов относится к траекторным базам данных, содержащим гео-позиционную информацию. Мы рассматриваем модель контекстно-зависимых вычислений (контекстно-зависимый браузер), которая основана на сетевой близости. Мобильный телефон рассматривается как сенсор близости и гео-позиционная информация заменена сетевой близостью. Любой существующий или специально созданный сетевой узел может рассматриваться как сенсор присутствия, который открывает доступ к динамически определяемому сетевому контенту. Раскрытие контента зависит от набора правил, описывающих условия сетевой близости. Дан алгоритм вычисления траекторий в мобильных сетях на базе информации о сетевой близости.

Ключевые слова: местоположение; траектории; LBS; сетевая близость; конвой, *context-aware*

1. ВВЕДЕНИЕ

Тема статьи относится к услугам мобильной сети, связанным с местоположением (*Location-based service*, LBS). Это тип информационных и развлекательных услуг, основанных на определении текущего местоположения мобильного телефона пользователя. Мы используем определение местоположения посредством точек доступа Wi-Fi, т.е. в окрестностях 20-30 м от точек доступа. Предложенная модель контекстно-зависимых вычислений (контекстно-зависимый браузер) основана на сетевой близости. Мобильный телефон рассматривается как сенсор близости и гео-позиционная информация заменена сетевой близостью.

Термин траектория представляет собой традиционную характеристику модели перемещения объекта в пространстве. Траектория может быть представлена как набор упорядоченных по времени точек, характеризующих конкретную позицию (моментальное расположение) объекта. Например, для мобильных телефонов (то, что и рассматривается в статье), это может быть временной ряд из координат, определенных с помощью GPS или CellId.

Под термином *конвой* подразумевается группа объектов, которые перемещаются совместно в течение некоторого заданного времени. Скорее всего, задача определения конвоев по данным траекторных измерений пришла из военных приложений. Вместе с тем, наибольший интерес представляют вполне мирные приложения. Например, планирование общественного транспорта в зависимости от выделенных шаблонов в перемещениях потенциальных пассажиров, оценка пропускной способности дорожных сооружений. Сервисы по совместному использованию автомобилей (*car pooling*) могут просчитывать свои модели, опираясь на данные о совместных перемещениях потенциальных пользователей и т.д.

В статье мы будем следовать каноническому определению конвоя [1] и избегать ограничений на размер группы и время совместного передвижения. Конвой далее – это группа объектов произвольного размера, определяемая метриками, основанными на плотности соединения [2]. Такой подход позволяет нам рассматривать произвольные группы объектов. Конвой – это группа перемещающихся объектов, примерно сохраняющая свою плотность в течение некоторого заданного времени. Объекты находятся в плотной связи, если в течение заданного времени расстояние между их последовательными позициями не превышает некоторой заданной величины. Это совпадает с интуитивным понятием “двигающиеся совместно”. Как следует из определения, такая классификация зависит от времени совместного передвижения. Применительно к нашей модели мы будем рассматривать относительно короткоживущие группы.

Другой часто используемый термин – это перемещающийся кластер (или кластер подвижных объектов) [3]. Члены этой группы могут свободно покидать ее или присоединяться повторно. Иными словами, с течением времени меняется не только местоположение объектов, но и их набор (состав группы). С точки зрения формальной чистоты определений, не каждый перемещающийся кластер может быть классифицирован как конвой. Вместе с тем, во многих работах такое различие не отслеживается, а оба термина используются как синонимы. Некоторые авторы используют в этой связи термин динамический конвой [4]. Это означает, что на разных этапах конвой (группа объектов) может иметь разный размер. Другой интересный термин – это стая (*flock*). Это группа объектов, которые перемещаются одинаково внутри некоторого временного интервала. В любом случае, все определения описывают совместное (групповое) перемещение, а вся разница в определениях состоит в том, каким образом описываются отношения между объектами.

В данной статье мы исследуем специальный случай для извлечения информации о перемещающихся кластерах (конвоях). В первую очередь, наша база данных не содержит в явном виде геопозиционных данных. Мы работаем с контекстно-зависимым приложением, предназначенным для доставки локального контента мобильным пользователям. Это приложение SpotEx (впервые представлено в [5]). Оно основано на идеях сетевой близости. SpotEx позволяет задать логические правила (т.н. продукции), которые зависят от сетевой близости и определяют доступность контента для мобильных подписчиков. Каждое сработавшее правило определяет некоторый контент (например, HTML страницу) для абонента.

Также это приложение может записывать историю (лог) информации о сетевой близости. Это некоторый временной ряд, который будет содержать цифровые оценки (метрики) близости к

сетевым узлам. Этот ряд становится аналогом траекторной базы данных. Например, при перемещении пользователя (в данном случае, мобильного телефона) в помещении приложение сохранит временной ряд для метрик сетевой близости.

Далее в статье приведена структура лога с данными о сетевой близости, а также способ получения измерений, что потребуют модификации классического определения конвоя. В разделе 2 содержится анализ существующих подходов к определению конвоев в гео-базах. В разделе 3 мы описываем контекстно-зависимый браузер SpotEx. И в разделе 4 приводится собственно алгоритм определения конвоев для SpotEx.

2. ВЫЯВЛЕНИЕ КОНВОЕВ

Приведем базовые определения, необходимые для выявления конвоев. Выделим основные аспекты [6]:

- а) кластеризация объектов: как описать (определить) кластер,
- б) постоянство состава: выделенная группа объектов должна сохранять состав в течение заданного времени,
- в) размер группы: многие прикладные задачи могут накладывать ограничения на размер конвоя, например, не меньше некоторой заданной величины.

Приведем базовые определения для задач выделения конвоев.

Соседство: для заданного порога расстояния e , множества точек S и оператора вычисления расстояния D соседство для точки p определяется как

$$NH_e(p) = \{q \in S \mid D(p, q) \leq e\}.$$

Близкая доступность: для заданного порога расстояния e и целого m , точка p прямо близко-доступна из точки q , если $p \in NH_e(q)$ и $|NH_e(q)| \geq m$. Точка p близко-доступна из точки q для заданного порога e и целого m , если существует последовательность точек p_1, p_2, \dots, p_n в множестве S такая, что $p_1 = q$, $p_n = p$, и p_{i+1} прямо близко-доступна из точки p_i .

Плотное соединение: дано множество точек S , точка $p \in S$ плотно соединена с точкой $q \in S$ для заданного порога e и целого m , если существует точка $x \in S$ такая, что обе точки p и q близко-доступны из x .

Определение плотного соединения является базовым для введения понятия конвоя. Это иллюстрирует рисунок 1.

Основываясь на плотном соединении объектов в течение некоторого заданного времени, конвой может быть определен следующим образом. Если дано множество траекторий N объектов, дан порог расстояния e , целое число m , целое количество временных отсчетов t , то конвой описывает группу объектов, состоящую из максимального числа близко-доступных объектов для заданных e и m в течение, по крайней мере, t временных отсчетов [1].

Комментарий к определению конвоя. Конвой можно определить как путешествующую компанию [6]. Группа объектов формирует компанию, если члены группы взаимно близко-доступны в течение некоторого заданного времени и размер группы не опускается ниже некоторого заданного порога.

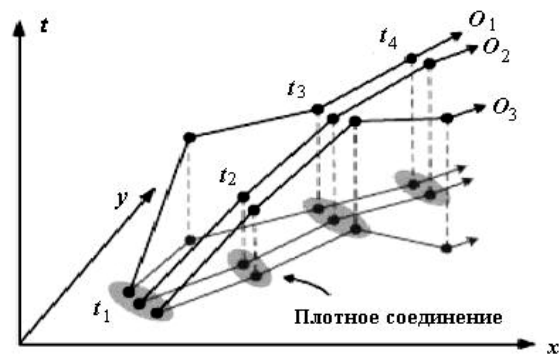


Рис.1. Пример конвоя [1].

Для определения путешествующих групп большинство алгоритмов использует концепцию кластера близкой доступности [7]. Простейшая и наиболее часто используемая техника для выделения конвоев состоит в последовательном выявлении кластеров близкой доступности с последующим определением общих объектов. На практике некоторые временные отсчеты для различных траекторий будут пропущены. Некоторые работы предлагают использовать линейную интерполяцию для создания виртуальных точек, относящихся к пропущенным временным отсчетам (см., например, алгоритм *Coherent Moving Cluster* [1]).

3. SPOT EXPERT

Что такое SpotEx

Изначально основная идея сервиса Spot Expert (SpotEx) была представлена как расширение позиционирования в помещениях на основе точек доступа Wi-Fi [5]. С течением времени проект SpotEx мигрировал к его текущему состоянию – к построению контекстно-зависимого браузера. Суть SpotEx состоит из двух частей.

Во-первых, SpotEx использует только часть процесса любой системы позиционирования, основанной на Wi-Fi, а именно – фазу определения доступных Wi-Fi сетей. В силу локальной природы радио-интерфейсов Wi-Fi, такое определение уже само по себе дает некоторую информацию о местоположении, то есть мы получаем информацию об относительном местоположении, о местоположении (близости) относительно некоторой (некоторых) точек доступа. Местоположение самих точек доступа при этом может быть и неизвестно. Собственно говоря, информация о местоположении в классическом смысле (например, широта и долгота места) никак не используется в SpotEx. Весь процесс построен на анализе близости.

Во-вторых (и это есть в основном), SpotEx использует некоторую внешнюю базу данных, содержащую правила (*если-то* операторы), относящиеся к конкретным точкам доступа. Основываясь на такого рода правилах, мы можем построить контекстно-зависимую систему извлечения данных и предоставлять контекстно-зависимые сообщения мобильным пользователям. Иными словами, в SpotEx доступность контента зависит от сетевого контекста – цифровой меры (отпечатка), представляющей сетевое окружение Wi-Fi.

Впервые SpotEx сервис [8], предложенный Д.Намиотом, был представлен на конференции NGMAST-2011 [5]. Текущее состояние сервиса, а также перспективы его развития представлены в работах [9] и [10].

SpotEx и Wi-Fi

Модель, используемая в SpotEx, позволяет избежать одной из самых главных (по крайней мере, в плане затрат) стадий подготовки систем позиционирования на базе Wi-Fi – разметки исходного пространства. Известные ныне системы позиционирования на основе Wi-Fi имеют существенный недостаток. Они базируются на подготовленной базе цифровых отпечатков сетевого окружения. Именно с такими, заранее подготовленными данными и сравнивается текущее сетевое окружение мобильного телефона. Естественно, что процесс подготовки такого рода баз данных достаточно трудоемкий. К тому же, его необходимо будет повторять, как только что-то меняется в сетевой структуре. При статической базе данных нет возможности задействовать динамические точки доступа Wi-Fi, например, когда точка доступа открыта непосредственно на мобильном телефоне.

В SpotEx местоположение подменяется близостью. Производственные правила, основанные на близости к элементам инфраструктуры Wi-Fi, включая и динамические элементы, заменяют информацию о точном позиционировании. Точки доступа Wi-Fi работают при этом как сенсоры присутствия. SpotEx не требует от мобильных абонентов присоединяться к Wi-Fi сетям, используемым для анализа. SpotEx использует только открытые (транслируемые) идентификацию сети (SSID) и другую публично-доступную информацию.

Основные компоненты SpotEx

1) *Серверная инфраструктура.* Сюда включена база данных с продукциями (правилами), программная компонента, отвечающая за механизм вывода на базе правил, и редактор правил. Редактор правил представляет собой веб-приложение (включая мобильный веб), которое обеспечивает пользовательский интерфейс к базе правил.

2) *Мобильное приложение.* Эта компонента отвечает за определение контекстной информации, сопоставление ее с существующими продукциями (правилами) и визуализацию вывода.

Достоинством сервиса SpotEx является то, что он может быть запущен на любой существующей или специально созданной Wi-Fi сети. Запуск системы не требует каких-либо изменений инфраструктуры. Редактор правил позволяет просто задать (описать) видимость контента внутри такой сети. Контент здесь есть просто некоторый текст (HTML код), который должен быть открыт на мобильном терминале при срабатывании соответствующего правила. Например, как только некоторая точка доступа становится “видима” мобильным телефоном, пользователю может быть открыт (показан) соответствующий контент.

Применения. Существующие модели применения связаны, в первую очередь, с местным маркетингом. Процесс полностью выглядит как автоматический “check-in” (отметка о присутствии) по аналогии с таким же сервисом в сети Foursquare. Это достаточно очевидная схема, например, магазин открывает доступ к маркетинговым материалам (например, купонам) для покупателей, которые оказались около некоторой точки доступа Wi-Fi.

При этом от мобильного абонента не требуется ни в какой форме непосредственно или с помощью некоторого API (программных интерфейсов) отмечать свое местоположение (подобно тому, как это делается в Foursquare, Facebook Places и т.д.). Именно такая отметка о присутствии в традиционных системах является основанием для предоставления купонов, скидок и т.п. В случае SpotEx мобильные подписчики могут собирать подобную информацию автоматически и без авторизации.

Другими перспективными областями, по нашему мнению, являются кампусы (распространение новостей и объявлений), а также Smart City проекты.

Один из наиболее интересных подходов к применению SpotEx состоит в том, что точка доступа Wi-Fi, которая используется в правилах раскрытия контента, может быть создана или открыта непосредственно на мобильном телефоне. Большинство современных смартфонов это позволяет. Мы можем привязать наши правила раскрытия контента именно к такой мобильной точке доступа, то есть контентные правила становятся привязанными к телефону.

Другой канонический пример мобильной точки доступа – автомобиль. Это так называемый *connected car*. В этом случае SpotEx поддерживает и модели, характерные для динамических LBS (*location based systems*). Доступные сервисы (контент в данном случае) будут автоматически “перемещаться” вместе с телефоном (мобильной точкой доступа), а также автоматически “включаться” и “выключаться” вместе с ней. Сервисы в данном случае могут автоматически следовать за персонами, которые их предоставляют.

Важно отметить, что наличие лишь одного смартфона достаточно для открытия нового информационного канала. Непосредственно на смартфоне с помощью мобильного веб-редактора может быть создан контент, здесь же могут быть описаны правила для его представления. Здесь же может быть открыта точка доступа Wi-Fi, к которой эти правила и привязаны.

Особенности SpotEx

Наш подход никак не затрагивает вопросы безопасности и присоединения к сети. Сервис SpotEx использует атрибуты (параметры) существующих точек доступа для раскрытия данных.

Каждое правило в SpotEx есть логическая продукция (*if-then* оператор). Другими словами, мы имеем набор логически правил, например:

```
IF IS_VISIBLE('mycafe') AND FIRST_VISIT() THEN
  {present the coupon info}
```

Условие (левая часть продукции) есть набор предикатов, оперирующих следующими данными (которые могут быть измерены/получены с помощью мобильного телефона):

- Идентификация Wi-Fi сети (SSID, MAC-адрес)
- RSSI (сила сигнала – дополнительно)
- Время дня (дополнительно)
- ID клиента (mac-address)

Заключение (в данном примере обозначенное как *{present the coupon info}*) представляет собой тот контент, который становится доступен мобильному пользователю в случае срабатывания данного правила. Это может быть, например, ссылка на некоторую мобильную веб-страницу, которая и открывается в браузере мобильного абонента.

На рисунке 2 приведен пример использования системы для маркетинга в ритейле.

Поскольку совокупность заданных правил представляет собой стандартную производственную систему, для ее реализации мы можем использовать хорошо известный алгоритм Rete [11].



Рис.2. SpotEx в сервисах, основанных на близости.

Реализация SpotEx

Как отмечено выше, каждое правило представляет собой продукцию (*если-то* оператор). Условия срабатывания правила зависят от указанных выше измерений и логических функций, определенных над ними. К числу таких функций (предикатов) в текущей версии реализации SpotEx относятся:

IS_VISIBLE ()
 NOT_VISIBLE ()
 CLOSE_THAN()
 FIRST_VISIT()
 FOLLOW_UP_VISIT()
 TIME()
 TIME_WITHIN()

Функция IS_VISIBLE() или NOT_VISIBLE() принимает в качестве параметра сетевой идентификатор (например, SSID или MAC-адрес для точки доступа) и возвращает булево значение в зависимости от видимости (доступности) указанной сети.

Функция CLOSE_THAN() в качестве параметров получает два идентификатора Wi-Fi сетей (точек доступа Wi-Fi) и возвращает значение ИСТИНА, если мобильный терминал ближе к первой точке доступа и ЛОЖЬ – в противном случае.

Две функции FIRST_VISIT() и FOLLOW_UP_VISIT() используют тот факт, что для Wi-Fi коммуникаций мы можем получить MAC-адрес клиента.

Накопление данных

Сама система SpotEx не требует авторизации от клиента, и контекстно-зависимый браузер позволяет просматривать данные анонимно. Но в то же самое время, в системе есть аналог глобального UUID, который может быть использован (с некоторым приближением, естественно) для различения пользователей. Это есть MAC-адрес. Мы можем сохранять лог элементов, описывающих беспроводной контекст (MAC-адрес, информация о сетевом окружении) и использовать его

для определения новых или повторных “посетителей”. Например, если для одного MAC-адреса у нас есть, по крайней мере, две исторические записи, то это вернувшийся посетитель. Это и есть базовый элемент для последующего анализа траекторий.

Используя журнал накопленных “снимков” сетевого окружения, мы и будем пытаться восстанавливать траектории. Зачем это нужно? С помощью анализа траекторий мы хотим узнать, каким образом мобильный абонент достиг данной точки внутри здания. Эти знания позволят нам определять новые правила для представления контента. Правила, которые будут зависеть еще и от траекторий. Например, кафе сможет предложить дополнительную скидку посетителям, пришедшим в составе некоторой группы и т.д.

SpotEx vs. гео-позиционирование

В современных гео-позиционных системах все механизмы оперирует понятием гео-отметка (или *check-in*). Все бизнес-процессы построены вокруг именно этого понятия и соответствующих ему действий. При этом практически полностью отсутствует история перемещений. Это особенно заметно на микро-уровне (для помещений). Предположим, что у нас есть новый *check-in* для Foursquare. Каким образом пользователь попал в отмеченное место? При обычном просмотре сайтов у нас есть поле *referrer* в заголовке HTTP запроса, которое может указывать на предыдущий сайт. Для гео-позиционных систем такого аналога нет. В этой работе предпринята попытка заполнить этот пробел. На практике это означает, что к указанному выше списку предикатов мы хотим добавить новую функцию:

$$\text{IN_GROUP_OF}(n, t)$$

Здесь n представляет некоторое положительное целое значение, а t описывает время (например, в секундах). Эта функция возвращает значение *true*, если мобильный пользователь передвигался в группе, по крайней мере, из n человек в течение, по крайней мере, t секунд. Естественно, это базируется на данных SpotEx. Иными словами, мы говорим только о тех мобильных пользователях, для которых есть записи в логе системы. То есть для каждого из упомянутых n пользователей присутствуют записи в системном логе. Мы считаем, что такая функция (в действительности – еще одна характеристика контекста) будет полезна для задач местного маркетинга, для извлечения данных в Smart City проектах и т.д.

Обсуждение SpotEx

Наша процедура анализа траекторий базируется на записях о сетевом окружении (цифровых снимках сетевого контекста). SpotEx начинает собирать записи с момента запуска приложения пользователем. Конечно, можно анализировать и исторически накопленные логи, но это отдельная задача, чему посвящен, например, проект Reality Mining [12].

Известна система Funf [13], которая также эксплуатирует модель “телефон как сенсор”. Мы тестировали возможность ее использования в проекте SpotEx. Funf Probes представляют собой базовые коллекции данных, собранных с помощью Funf framework. Каждый такой объект отвечает за сбор определенного типа данных. В системе присутствуют объекты, которые отвечают и за сбор данных от мобильных сенсоров, реализованных в телефоне. Таких, например, как акселерометр, GPS и т.д. Сюда же, естественно, входят и сенсоры для беспроводных сетей. Это весьма развитый инструмент для сбора контекстной информации на телефонах Android. Для анализа траекторий в проекте SpotEx мы используем малую часть того, что может быть собрано с помощью Funf.

4. ТРАЕКТОРИИ В ЛОГЕ СЕТЕВОЙ ИНФОРМАЦИИ.

Предлагаемый нами контекстно-зависимый браузер собирает информацию о сетевом окружении в процессе работы. Точнее, программа собирает набор “снимков”, каждый из которых описывает состояние сетевого окружения (имеются в виду Wi-Fi сети) в определенный момент времени. Фиксация такого рода информации есть, например, стандартная функция систем позиционирования, основанных на Wi-Fi. Каждая запись о сетевом окружении представляет собой привязанный к некоторому моменту времени список записей. Каждая запись, в свою очередь, есть вектор триплетов. Каждый триплет описывает одну Wi-Fi сеть, которая достижима с конкретного мобильного телефона:

Network ID (SSID)
MAC-адрес
Сила сигнала (RSSI)

и все окружение может быть описано как вектор триплетов:

$$E = \{T_1, T_2, \dots, T_n\}.$$

“Снимок” сетевого окружения (именно для него используется термин *fingerprint* – отпечаток) – это указанный выше вектор, снабженный временной меткой: $[t_i, E_i]$.

О пропущенных измерениях

Окончательно, мы получаем некоторый временной ряд. Отметим, что технически наша регистрирующая программа (основанная на SpotEx или Funf) производит замеры с регулярным временным интервалом. Но, исходя из тех же самых технических соображений, не все полученные данные могут быть доступны в момент вычисления функции `IN_GROUP_OF()`. В первую очередь, для экономии питания на мобильном телефоне, регистратор может кэшировать данные о сетевом окружении и сохранять их на центральном сервере (в общем логе, где они и анализируются) в пакетном режиме, например, сохранять данные только на каждом третьем (пятом и т.д.) цикле опроса. Естественно, при анализе возникает важный вопрос о пропущенных данных. Это общая проблема при восстановлении траекторий. И, естественно, возникает вопрос об устойчивости алгоритмов по отношению к проблеме такого рода. Анализ литературы показывает несколько подходов к решению проблемы пропущенных данных. Одно из возможных решений – введение некоторого периода неактивности для кандидатов в члены конвоя (группы). Этот период неактивности задает максимально допустимый порог времени между двумя позициями, относящимися к одному и тому же объекту. Для каждого объекта с пропущенными измерениями мы сравниваем время последнего известного измерения и период неактивности. Если период еще не истек, то объект считается относящимся к тому же кластеру, к которому он был отнесен на момент последнего измерения. Иными словами – мы считаем, что он продолжает путешествие с той же самой компанией. В нашей текущей реализации пропущенные значения просто игнорируются и соответствующие объекты (мобильные телефоны) удаляются из групп. Другие стратегии, равно как и анализ устойчивости, мы оставили до следующих работ.

О понятии близости

Другой важный момент, который необходимо отметить, связан с общими принципами измерений, основанных на близости. Предположим, что наша точка доступа Wi-Fi имеет антенну с круговой диаграммой направленности. Как показано на рис.3, базируясь только на информации о близости, мы не сможем понять, что группы 1 и 2 двигаются к центру с противоположных сторон.

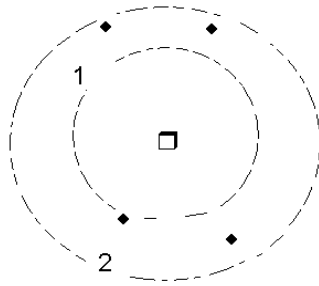


Рис.3. Wi-Fi антенна с круговой диаграммой направленности.

Если группы 1 и 2 двигаются как-то согласованно, то анализируя только близость к точке доступа (показана в центре) их можно формально отнести в один кластер. В задачах SpotEx такие группы будем рассматривать как одну. Но это означает, что определение конвоя (путешествующей группы) нужно переформулировать. По существу, нас интересует согласованное перемещение мобильных абонентов вблизи элементов сетевой инфраструктуры (точки доступа), и совместно путешествующей группой будем считать группу объектов (мобильных телефонов, на практике), которые имеют схожие характеристики записей о сетевом окружении внутри заданного временного интервала. Будем считать две последовательности сетевых “снимков” подобными на некотором временном интервале, если для каждого последовательного измерения в первом временном ряду (последовательности) может быть найден последовательный замер во втором ряду, примерно в то же самое время и со сравнимыми сетевыми характеристиками. Сетевые характеристики (снимки) будут считаться сравнимыми, если есть хотя бы одна пара сравнимых триплетов.

Рассмотрим это на примере. Предположим, что у нас есть два временных ряда:

$$T_1 \text{ as } \{[t_{11}, E_{11}], [t_{12}, E_{12}], [t_{13}, E_{13}], \dots\} \text{ and } T_2 \text{ as } \{[t_{21}, E_{21}], [t_{22}, E_{22}], [t_{23}, E_{23}], \dots\}$$

Здесь t_{ij} описывает временную метку, а E_{ij} описывает Wi-Fi окружение. Схожесть (подобие) обозначает, что мы можем отобразить измерения из первого ряда во второй (найти пару элементам первого временного ряда). При этом наше отображение не должно нарушать временной последовательности. То есть, если мы, например, отобразили точку $[t_{11}, E_{11}]$ в $\{[t_{21}, E_{21}]\}$, то в паре для следующей точки $[t_{12}, E_{12}]$ мы должны видеть время $t \geq t_{21}$.

О точности измерений

Поскольку каждое приложение (каждый мобильный телефон) собирает данные независимо, мы не можем гарантировать, что для произвольной временной метки в первом ряду найдется измерение, сделанное в то же самое время во втором ряду. Для произвольной временной метки t_{1i} может не существовать точно такого же временного отсчета t_{2j} . Поэтому в определении и говорится о “приблизительно таком же времени”. Мы будем искать соответствия в интервале $t_{1i} \pm \Delta$, где Δ есть некоторый постоянно выбранный порог (погрешность). Выбор конкретного значения для этого порога может быть, конечно, связан с регулярным интервалом сбора измерений. Иными словами, если мы будем сравнивать этот подход с традиционными задачами извлечения знаний из траекторной информации, то можно заметить, что мы не пытаемся восстановить (предсказать или аппроксимировать) пропущенные значения. Если мы не можем найти какого-либо соответствия одному измерению в другом временном ряду внутри заданного интервала, мы просто полагаем эти два временных ряда несравнимыми.

Сравнение силы сигнала

Исходя из тех же причин (каждый телефон собирает данные независимо от других участников), мы не можем гарантировать, что на двух телефонах будут получены абсолютно совпадающие значения измерений Wi-Fi сигнала для одного и того же места. Это зависит от антенны телефона, уровня заряда батареи, внешних условий и т.д. Тем самым, для сравнения сетевых измерений появляется еще одно пороговое значение, которое описывает максимально допустимое расхождение в силе сигнала. При сравнении триплетов измерений идентификация сети должна быть одинакова, а значения замеров для силы сигнала могут различаться не более чем на некоторую фиксированную величину. Поэтому мы говорим о “приблизительно совпадающих” Wi-Fi измерениях.

Два сетевых измерения сравнимы в данной статье (и это простейшая метрика), если они имеют, по крайней мере, одну общую точку доступа с разницей в силе сигнала не более чем некоторое фиксированное значение.

Алгоритм вычислений

Наша задача состоит в вычислении логической функции $IN_GROUP_OF()$. Эта функция вызывается внутри одного из предикатов контекстного правила, проверяемого для какого-то мобильного пользователя. Это означает, что как исходную точку мы можем использовать измерения сетевого окружения для данного мобильного пользователя (“снимок” Wi-Fi окружения для мобильного пользователя, в интересах которого и вычисляется эта функция).

Начальные параметры:

Δ – временной порог, Ω – RSSI порог (порог для силы сигнала), E – исходное сетевое окружение, T_0 – начальное время, T_{max} – аргумент функции – время жизни группы.

1. Создаем пустое множество R_1 ;
2. Собираем в него измерения на временном интервале $T_0 - \Delta \rightarrow R_1$;
3. **Если** R_1 пусто, **то** возвращаем *false*;
4. Удаляем из R_1 все измерения, несравнимые с E ;
5. **Если** R_1 пусто, **то** возвращаем *false*;
6. **Set** $t = T_0$;
7. **While** $t > T_0 - T_{max}$
 8. Находим предыдущее измерение для текущего пользователя.
Обновляем значения $\rightarrow \{t, E\}$;
 9. Для каждого измерения в R_1 находим новый отсчет внутри интервала $t \pm \Delta$ (обновляем измерения новыми данными);
 10. Удаляем из R_1 элементы без новых данных (не обновленные);
 11. Удаляем из R_1 элементы несравнимые с E ;
 12. **Если** R_1 пусто, **то** прерываем цикл;
13. **End while**

Это алгоритм кластеризации, примененный к “снимкам” сетевого окружения. Окончательно, множество R_1 представляет искомую группу. В зависимости от размера этого множества мы и можем вычислить логическое значение для функции $IN_GROUP_OF()$.

О дальнейших работах

Результаты работы будут использоваться для расширения функционала нового сервиса для контекстно-зависимого извлечения данных. Будет проведен анализ устойчивости алгоритма, получены численные данные о скорости работы и точности классификации.

5. ЗАКЛЮЧЕНИЕ

В статье рассмотрено новое приложение (новая модель использования) для задач выявления конвоев в траекторной информации. При этом известные подходы использованы для новой модели применения исходя из требований задачи контекстно-зависимых вычислений. Журнал с информацией о сетевом окружении используется здесь как замена классической базы данных о траекториях движения. Этот факт, а также особенности измерений и сбора информации потребовали изменения и модификации стандартных определений и алгоритмов, применяющихся для извлечения шаблонов из траекторных измерений.

СПИСОК ЛИТЕРАТУРЫ

[1] *H. Jeung, M. Yiu, X. Zhou, C.Jensen, and H.Shen.* Discovery of convoys in trajectory databases// Journal Proceedings of the VLDB Endowment, Vol.1, No.1, August 2008, pp.1068-1080.

[2] *M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.* A density-based algorithm for discovering clusters in large spatial databases with noise // SIGKDD, pp.226–231, 1996.

[3] *P. Kalnis, N. Mamoulis, and S. Bakiras.* On discovering moving clusters in spatio-temporal data// SSTD, pp.364–381, 2005.

[4] *H. Aung and K.-L. Tan.* Discovery of Evolving Convoys, Scientific and Statistical Database Management // Lecture Notes in Computer Science, 2010, Vol. 6187/2010, pp.196-213. DOI: 10.1007/978-3-642-13818-8_16.

[5] *D. Namiot and M. Schneps-Schneppe.* About location-aware mobile messages //International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST), 14-16 Sept. 2011, pp.48-53. DOI: 10.1109/NGMAST.2011.19.

[6] *L.Tang, Y. Zheng, J.Yuan, J.Han, A.Leung, C.Hung, and W.Peng.* On Discovery of Traveling Companions from Streaming Trajectories// Microsoft Research.

<http://research.microsoft.com/pubs/156047/On%20Discovery%20of%20Traveling%20Companions%20from%20Streaming%20Trajectories.pdf>. Retrieved: Aug, 2012.

[7] *M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.* A density-based algorithm for discovering clusters in large spatial databases with noise// SIGKDD, 1996.

[8] *Y. Daradkeh, D. Namiot, M. Sneps-Sneppe.* Spot Expert as Context-Aware Browsing // Journal of Wireless Networking and Communications, vol.2, No.3, 2012, pp.23-28. DOI: 10.5923/j.jwnc.20120203.03.

[9] *Y. Daradkeh, D.Namiot, and M.Sneps-Sneppe.* Context-Aware Browsing for Hyper-Local News Data // International Journal of Interactive Mobile Technologies (IJIM), vol.6, No.3, 2012, pp.13-17, DOI: 10.3991/ijim.v6i3.2053.

[10] *D.Namiot.* Context-Aware Browsing – A Practical Approach// Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on, pp.18-23. DOI: 10.1109/NGMAST.2012.13.

- [11] *E. Friedman-Hill*. Jess in action: rule-based systems in Java. Manning Publications Co. Greenwich, CT, USA 2003 ISBN: 9781930110892.
- [12] *A.Pentland, T.Choudhury, N.Eagle, and P.Singh*. Human Dynamics: Computation for Organizations // Pattern Recognition Letters, Vol. 26, No.4, 2004, pp.503-511.
- [13] Funf Open Sensing Framework. <http://funf.media.mit.edu/> <retrieved: Aug 2012>.

Рукопись получена 17.12.2012