

Глубокий анализ мобильного трафика и кластеризация базовых станций для оптимизации C-RAN

Задворный Даниэль

МГУ ВМК 217 группа

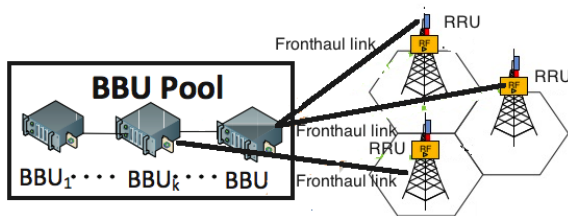
18 апреля 2023 г.

Outline

- 1 Прогнозирование
- 2 Терминология
- 3 Подготовка к кластеризации
- 4 Кластеризация с учетом ограничений на расстояния
- 5 Оценивание работы алгоритма
- 6 Результаты

Архитектура C-RAN

Что это такое и зачем нам предсказывать трафик



Определение

C-RAN (Cloud-RAN), также известная как Centralized-RAN, это сеть радиодоступа на основе облака, которая координирует работу многих базовых станций, в которых базовые блоки BBU виртуализованы в облаке.

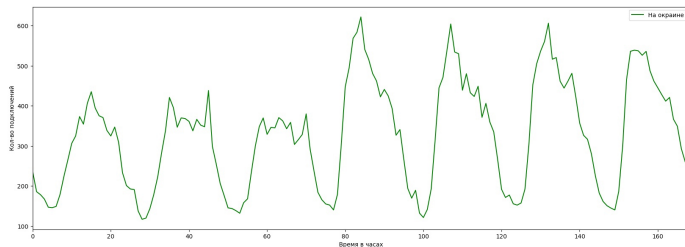
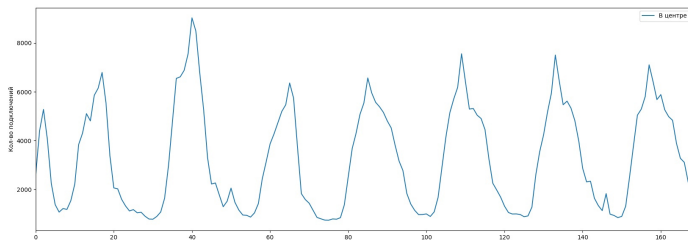
Прогнозирование

Какие методы прогнозирования были использованы?

1. ARIMA - Autoregressive Integrated Moving Average
2. WANN Windowed-Artificial Network модели

Сравним прогнозирование в деловом центре и на окраине города

Ко-во подключений в центре города



Базовая идея алгоритма

Нам необходимо точно прогнозировать трафик, учитывая пространственную и временную корреляцию между радио вышками, именно поэтому будет использоваться Recurrent Neural Network, которая автоматически будет находить временную и пространственную связь между объектами. Наш алгоритм будет основан на WANN. Считается, что обучение нейронной сети будет очень тяжелым из-за **Vanishing** или **Exploding Gradient** проблемы, поэтому авторы будут использовать **multivariate-Long Short-term Memory Network**

Терминология

RRH - Remote Radio Head

Удаленные радио точки могут быть описаны как такое множество:

$$\{r | r = (rid, lng, lat)\}$$

rid - id точки

lng, lat - координаты точки

RRH трафик

RRH трафик

Трафик собранный с каждой RRH, может быть представлен как множество фиксированной длины:

$$\{f_i | f_i = [u_i(1), \dots, u_i(t), \dots, u_i(N_t)]\}$$

$u_i(t)$ - это объём трафика i -ой RRH, с промежутком в промежутке времени t в нашем случае он равен 24 часам

$$(1 \leq t \leq N_t)$$

Матрица трафика

Матрица трафика

Пусть $F \in Mat(R_{N_t} \times R_{N_r})$, где N_t - кол-во промежутков времени, N_r - кол-во RRH в сети

Давайте обозначать $F([0, t], :)$ - трафик всей сети до времени t

Трафик сети, который хотим предсказать на время Δt - $F([t, t + \Delta t], :)$

RRH Traffic Snapshot

Снимок Трафика

Снимком трафика будем называть матрицу F_i , которая отвечает за трафик в течение заданного времени в период Δt

$$\mathcal{F} = \{F_i | F_i = F([(i-1) \cdot \Delta t, i \cdot \Delta t], :), i = 1, 2, \dots\}$$

Будем использовать Seq2Seq модель, дополняющую нашу LSTM модель.

Прогнозирование будет происходить так: модель принимает F_i и выдает F_{i+1}

Структура MuLSTM модели

Наконец мы добрались до структуры. В общем случае MuLSTM модель имеет два LSTM слоя L_1 и L_2 . Энкодер L_1 принимает снимок размера $\Delta t \times N_r$, изучает его внутреннюю структуру и в зашифрованном виде отправляет декодеру L_2 , который делает прогноз будущего снимка. Модель была натренирована посредством Backpropagation Through Time (BPTT) алгоритмом. $N_{encoder} = N_{decoder} = 32$, $N_{iter} = 10000$

Подготовка к кластеризации

Научились прогнозировать, что дальше?

Так как мы научились предсказывать трафик, то теперь мы можем посчитать насколько взаимосвязаны каждые RNN, для этого нам нужно ввести метрику, при этом мы хотим учесть два аспекта:

- Распределение пиков
- Загруженность

Распределение пиков

Пиковый объем трафика каждой RRH в кластере должен быть распределен равномерно в течение дня, для того чтобы объем VBU мог быть разделен между всеми RRH.

Пусть у нас есть кластер $C = \{r_1, \dots, r_n\}$, для начала найдем пиковые часы каждой точки в течение дня $T(r_i) = \{t_{i_1}, t_{i_2}, \dots, t_{i_m}\}$, где t_{i_j} - время наступления j -го пика, $1 \leq t_{i_j} \leq 24$.

Затем давайте посчитаем **энтропию Шенона** для пиковых часов для кластера C , $T(C) = \bigcup T(r_i)$

$$H(C) = - \sum_{k=1}^K p_k \cdot \log(p_k)$$

где $K = |T(C)|$, p_k - вероятность наблюдать соответствующий пиковый час. Чем выше энтропия тем больше взаимосвязей между RRH в кластере.

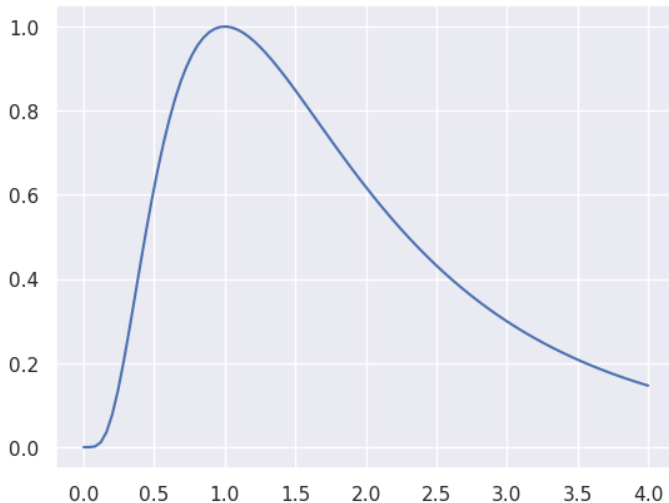
Загруженность

Чтобы использовать по максимуму преимущество нашей C-RAN архитектуры, емкость VBU соответствующая кластеру должна быть близка к агрегируемому кластером трафику в различные часы дня, но не должна быть слишком близка, чтобы уберечь его от перегрузки. Определим

$$U(C) = \left(\frac{\text{mean}f(C)}{|B|} \right)^{-\ln \frac{\text{mean}f(C)}{|B|}}$$

где $f(C) = \sum_{i=1}^n f(r_i)$ - трафик создаваемый кластером C , $|B|$ - фиксируемый объем VBU

Загруженность



Итоговая метрика для кластера

$$M(C) = U(C) \cdot H(C) = - \left(\frac{\text{mean}f(C)}{|B|} \right)^{-\ln \frac{\text{mean}f(C)}{|B|}} \cdot \sum_{k=1}^K p_k \cdot \log(p_k)$$

Взвешенный граф

Граф

Будем моделировать взаимосвязь между RRH как взвешенный не ориентированный граф $G = (V, E)$, где $V = \{r_1, \dots, r_N\}$, E - множество рёбер между двумя вершинами.

Матрица смежности

Матрица смежности A графа G размера $N \times N$, в которой $a_{ij} = 1$, если есть ребро между r_i и r_j , 0 иначе. В нашем случае будем говорить

$$a_{ij} = \begin{cases} 1 & \text{dist}(r_i, r_j) \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

τ - ограничение на географическое расстояние между точками.

$$w(r_i, r_j) = a_{ij} * M(\{r_i, r_j\})$$

Кластеризация с учетом ограничений на расстояния

Определение задачи

На этом шаге мы должны кластеризовать RRH's так, чтобы каждый кластер состоял из соседних RRH связанных между собой, с учетом весов. Эта задача может тогда называться **community detection problem**.

Дан граф $G = (V, E)$, для начала определим $\mathcal{P} = \{C_1, C_2, \dots, C_K\}$, т.ч.

$$\bigcup_{C_k \in \mathcal{P}} C_k = \mathcal{P} \text{ и } \bigcap_{\forall C_k \in \mathcal{P}} = \emptyset$$

- ▶ Определим связность RRH и кластера $con(v, C) = \sum_{v' \in C} w(v, v')$
- ▶ Определим смежный кластеры к RRH v $\mathbf{C}(v) = \{C | con(v, C) > 0, C \in \mathcal{P}\}$
- ▶ Тогда будем называть C_k оптимальным кластером, если $\forall v \in C_k$,
 $con(v, C_k) \geq \max\{con(v, C_l), C_l \in \mathcal{P}\}$

Причем $\forall v, v' \in C_k$ $dist(v, v') \leq \tau$

Решение

Будем пользоваться для кластеризации алгоритмом, который называется DCCA(**D**istance-**C**onstrained **C**omplementarity **A**ware)

Базовая идея

Мы жадно добавляем вершину к смежному кластеру, с наибольшим значением функции

$$value(v, C) = con(v, C) \times \log\left(\frac{\tau}{\max\{dist(v, v')\}}\right)$$

Так как сходимость такого алгоритма доказать тяжело установим максимальное кол-во итераций, после которого алгоритм будет остановлен.

Решение

Опишем алгоритм более подробно

- ▶ Инициализация: каждая вершина в графе - кластер
- ▶ На каждой итерации мы случайным образом генерируем список \mathbf{L}
 1. Итерируемся по \mathbf{L}
 2. Сначала удаляем RRH из текущего кластера и ищем смежные кластеры
 3. Считаем $value(,)$ для всех смежных кластеров и помещаем вершину в кластер с наибольшим значением функции
 4. помечаем вершину как *moved*, если кластер не совпадает с изначальным
- ▶ Решаем пойти ли на новую итерацию, критерии останова:
 1. Ни одна вершина не была помечена как *moved*
 2. Было достигнуто максимальное кол-во итераций *maxiter*

Оценивание работы алгоритма

Средняя абсолютная ошибка (MAE) для прогнозирования снимка трафика, \tilde{F}_i - прогноз, F_i - реальность

$$MAE(F_i, \tilde{F}_i) = \frac{\sum_{t=1}^{N_t} \sum_{r=1}^{N_r} |F_i(t, r) - \tilde{F}_i(t, r)|}{N_r \times N_t}$$

Во время работы нашего алгоритма подсчитывалось два аспекта:

- Увеличение средней заполненности VBU
- Уменьшение стоимости системы

Оценивание

Средняя заполненность

Для набора кластеров $\mathbf{P} = \{C_1, \dots, C_k\}$ определим метрику, основанную на предыдущей

$$Utility(\mathbf{P}) = mean_{C_k} U(C_k)$$

Стоимость системы

Для оценивания стоимости системы давайте посчитаем общую вместимость набора кластеров $\mathbf{P} = \{C_1, \dots, C_K\}$

$$Cost(\mathbf{P}) = \sum_{k=1}^K |\{C_k\}|$$

Результаты

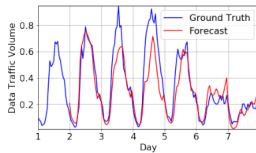
Какие методы были использованы?

- ▶ Традиционный
- ▶ ARIMA-DCCA
- ▶ WANN-DCCA
- ▶ MuLSTM-DC
- ▶ MuLSTM-DCCA

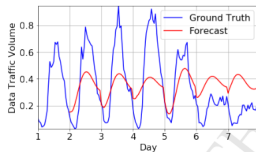
Table 2. Evaluation Results

Methods	Traffic Forecast Error (MAE)		Average Capacity Utility		Overall Deployment Cost	
	Milan	Trentino	Milan	Trentino	Milan	Trentino
Traditional	-	-	38.8%	29.4%	182	522
ARIMA-DCCA	0.202	0.237	65.3%	45.2%	112	160
WANN-DCCA	0.175	0.198	73.4%	58.8%	96	120
MuLSTM-DC	0.074	0.083	58.7%	39.2%	120	180
MuLSTM-DCCA (Proposed)	0.074	0.083	83.4%	76.7%	88	270

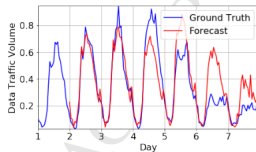
Визуализация прогнозов



(a) MuLSTM forecast

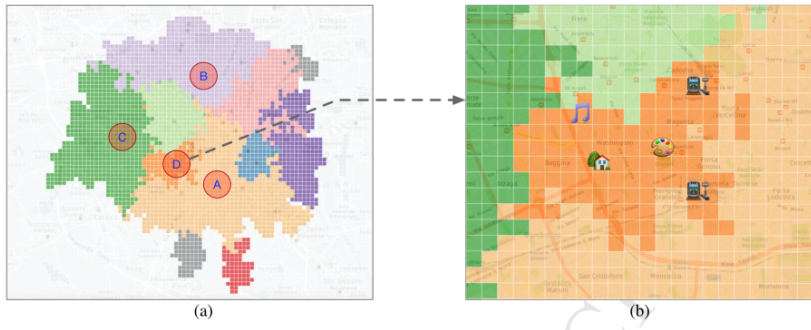


(b) ARIMA forecast



(c) WANN forecast

Изображение кластеризации



Изображение кластеризации

