Оценка матрицы транспорта по неполной информации

Olga Grebenkova
grebenkova.os@phystech.edu

Daniil Merkulov
daniil.merkulov@skoltech.ru

Project Proposal

Проект заключается в исследовании способов оценки матрицы транспорта(корреспонденций) в условии невозможности её прямого измерения. Конкретно будет исследованы способы, представленные в статьях [1, 2]: линейное программирование и использование случайной нейронной сети(англ. «Random Neural Network») соответсвенно.

1 Идея

1.1 Problem

Определение 1 *Матрица корреспонеденций(трафика) — матрица, отражающая поток между всеми возможными парами вершин в сети.*

Рассмотрим сеть с n вершинами. Тогда $c = n(n-1)$ - количество всех возможных пар источник-приёмник в графе (мы не расматриваем путь из вершины в саму себя). Пусть r общее количество рёбер в сети.

Для любой упорядоченной пары вершин $a = (i, j)$ назовём X_a количеством сообщений (потоком) из вершины i в вершину j. Для любой упорядоченной пары $s = (i, j)$, соединенной ребром, назовём Y_s поток по ребру $i \to j$.

Определение 2 *Поток по всем рёбрам* ($\mathbf{Y} = (Y_1, \dots, Y_r)^\top$) *— вектор, содержащий в качестве компонент поток по каждому из рёбер.*

Также рассмотрим векторное представление матрицы корреспонденций: $\mathbf{X} = (X_1, \dots, X_c)^\top$.

Обычно, измерять напрямую $\mathbf{X}$ очень затратно, так как количество всевозможных пар c огромно. Поэтому во всех методах оценки используют $\mathbf{Y}$, так как его измерение представляется возможным и незатратным с помощью различных сетевых протоколов. В дальнейшем в этой работе будем считать, что $r < c$.

Определение 3 *Матрица пути* $\mathbf{A}$ *— матрица, размерами* $r \times c$*, где* $A_{s,a} = 1$*, если ребро* $s = (i, j)$ *принадлежит пути между парой вершин* $a = (k, l)$*. Путь будем считать построенным по алгоритму Дейкстры с использованием потока по ребру в качестве его веса.*

Тогда можем записать отношение матриц корреспонденций к потоку по всем рёбрам:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{1}$$

1.2 Линейное программирование

В [1] описана постановка задачи линейного программирования для решения данной проблемы.

Так как Y_s равняется сумме всех потоков проходящих через ребро s, можем поставить следующую задачу оптимизации:

$$\max \sum_{a=1}^{c} w_a X_a, \tag{2}$$

где w_j - вес для пары j. Зависимость результата от веса w_j так же можно исследовать. Некоторые результаты представлены в [1].

При этом накладываются следующие ограничения:

$$\sum_{a=1}^{c} A_{s,a} X_a \leq Y_s \quad s = 1, \dots, r$$

$$X_a \geq 0 \quad \forall a$$

$$\sum_{s=(i,j)} Y_s A_{s,k} - \sum_{s=(j,i)} A_{s,k} = \begin{cases} X_k & j \text{ источник для } k \\ -X_k & i \text{ конечная точка } k \\ 0 & \text{иначе} \end{cases}$$

1.3 Случайная нейронная сеть

В [2] представлен алгоритм решения задачи, использующий случайную нейронную сеть.

Определение 4 *Случайная нейронная сеть(RNN) — математическая модель, представляющаяся собой смесь классической нейронной сети и теории очередей.*

RNN (так же как и в классическом случае) представляет собой набор взаимосвязанных нейронов. Каждый из них обменивается импульсами с внешней средой и другими нейронами и имеет потенциал - положительное целое число.

Потенциал нейрона i в момент времени t обозанчим как $q_t(i)$. Если $q_t(i) > 0$ — считаем нейрон возбужденным. В этом состоянии он случайно посылает сигнал в соответствии с пуассоновским случайным процессом с интенсивностью r_i. В данной модели нейроны могут обмениваться как положительными, так и отрицательными сигналами. Вероятность того, что сигнал, отправленный нейроном i, пойдёт в нейрон j со знаком «+» или «-» обозначим через $p^+_{i,j}/p^-_{i,j}$. Сигнал покидает сеть с вероятностью d_i. Если нейрон получает положительный сигнал, его потенциал увеличивается на 1. Если он получает отрицательный сигнал или нейрон отсылает сигнал, то его потенциал уменьшается на 1. Наименьшее значение потенциала — 0. Поток положительных и отрицательных сигналов из внешней среды к нейронам также соответствуют пуассоновским процессам с интенсивностью λ^+_i и λ^-_i соответственно. Наконец, вместо использования вероятностей $p^+_{i,j}/p^-_{i,j}$, будем использовать веса нейронной сети $w^+_{i,j} = r_i p^+_{i,j}$ и $w^-_{i,j} = r_i p^-_{i,j}$.

Основная идея метода состоит в том, чтобы найти некоторое нелинейное отображение $f_k(\cdot) : \mathbb{R}^{n^k} \to \mathbb{R}$ для каждого потока между парой k в момент времени — $X_t(k)$:

$$X_t(k) = f_k(\mathbf{Y_t}(\delta) = f_k(Y_t(\delta^1_k), Y_t(\delta^2_k), \dots, Y_t(\delta^{n_k}_k)) \tag{3}$$

где $Y_t(\delta_\mathbf{k})$ — вектор, содержащий поток через все n_k рёбер, через которые проходит путь между парой вершин k, а $\delta_k = (\delta^1_k, \delta^2_k, \dots, \delta^{n_k}_k)$ - вектор состоящий из n_k ненулевых значений столбца k матрицы пути.

Таким образом, нелинейное отбражение $f_k(\cdot)$ "получает" значение потока между парой вершин k, используя путь между ними. Вычисление этой функции можно расценивать как вычисление псевдообратной матрицы из матрицы пути для конкретного элемента матрицы корреспонденций. Следующим шагом алгоритма является поределение независимых отображений $f_k(\cdot)$ — по одному для каждой пары X_k, используя RNN модель.

2 Outcomes

В качестве результата проекта планируется предоставить постер с анализом приведенных выше алгоритмов на основании графика точности предсказания матрицы корреспонденций от количества эпох, весов в линейной модели, интенсивности случайных процессов и прочих параметров моделей.

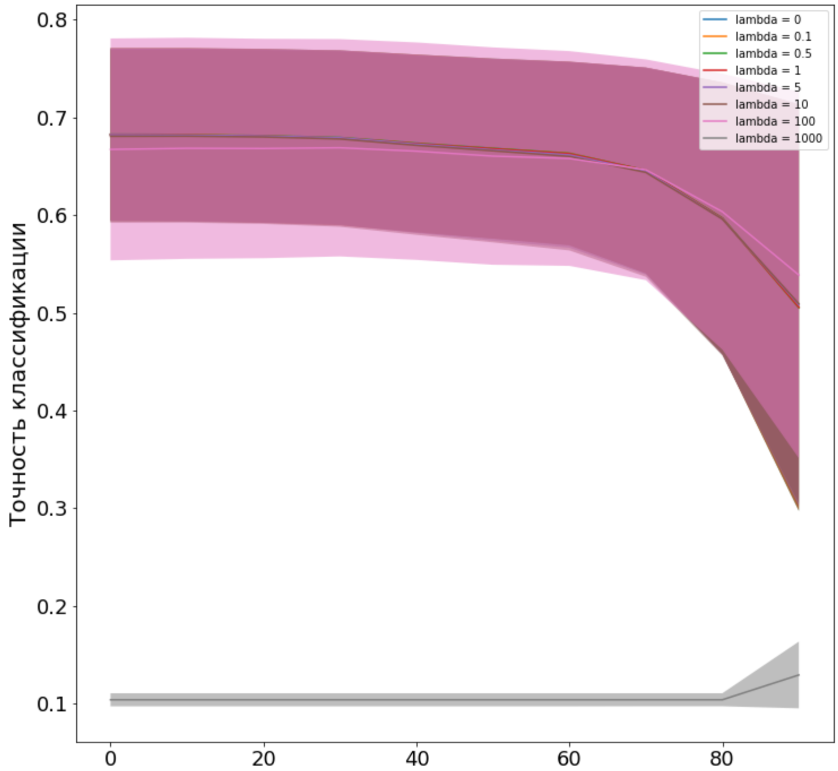

Figure 1: Зависимость точности определения матрицы от интенсивности пуассоновского процесса входных сигналов, где λ - интенсивность пуассоновских процессов распространения сигнала

3 Литературный обзор

В работе [1] были исследованы следующие три способа оценки матрицы транспорта: линейное программирование, байессовский подход и EM-алгоритм. Основной работой, в которой был предложен байессовский подход к решению данной задачи, является работа [3]. При этом подходе предполагается, что $X_j \sim Poisson(\lambda_j)$, то есть имеет пуассоновское распределение. В работе [4] описываются возможные применения EM-алгоритма. Данная работа дополнена [5], где представлено применение EM-алгоритм к нахождению матрицы корреспонденций. В [1], авторы показали, что основные предположения, лежащие в основе этих статистических моделей не всегда оправданы, и что эти методы плохо работали, когда основные предположения, такие как пуассоновское распределение потока, нарушались.

В дальнейших работах в решениях задачи были учтены особенности пространства, что привело к графитационной модели [6].

В последние годы к решению проблемы матрицы корреспонденций стали активно применять нейронные сети. В работе [7] используется трёх-слойная модель классической нейронной сети. Метод прдложенный в этой работе работает лучше классических, изложенных раннее, но имеет один существенный недостаток: результат работы сильно зависит от конкретной реализации нейронной сети и количества нейронов.

В работе [2] представлен метод использующий случайные нейронные сети, впервые представленные в работе [8]. Эксперименты показывают, что данный метод работает стабильнее метода из [7] и имеет меньшую зависимость от топологии нейронной сети.

4 Метрики качества

Оценить результат работы можно будет по точности работы алгоритма на соответствующих выборках. Еще одним показателем может быть быстродействие и функция ошибки.

5 Примерный план

- Во вторник узнать побольше про линейное программирование и в выходные 11.04 - 12.04 попробовать написать код для линейного алгоритма. Жесткий дедлайн: 20.04.20 00:00
- Построить графики для линейного случая до 20.04.20 00:00
- Разобраться до конца в методе случайных сетей. Написать сетку. Мягкий дедлайн: 28.04.20. Жесткий: 04.05.20
- Построить графики сравнения первого и второго метода до 12.04.20 23:59
- Программа минимум -анализ задачи линейного программирования. Если не осилю сделать до жесткого дедлайна оставляю, только ее.
- Программа максимум - анализ решения через рандомную нейронную сеть.

References

[1] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Computer Communication Review*, volume 32, pages 161–174, oct 2002.

[2] Pedro Casas and Sandrine Vaton. On the use of random neural networks for traffic matrix estimation in large-scale IP networks. In *IWCMC 2010 - Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, pages 326–330, 2010.

[3] Claudia Tebaldi and Mike West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–573, jun 1998.

[4] Shu Kay Ng, Thriyambakam Krishnan, and Geoffrey J McLachlan. The EM algorithm. In *Handbook of Computational Statistics: Concepts and Methods: Second Edition*, pages 139–172. Springer Berlin Heidelberg, 2012.

[5] Jin Cao, Drew Davis, Scott Vander Wiel, Bin Yu, and Bin Yu. Time-Varying Network Tomography: Router Link Data. *Journal of the American Statistical Association*, 95(452):1063–1075, dec 2000.

[6] Matthew Roughan, Albert Greenberg, Charles Kalmanek, Michael Rumsewicz, Jennifer Yates, and Yin Zhang. Experience in Measuring Backbone Traffic Variability: Models, Metrics, Measurements and Meaning. In *Proceedings of the 2nd Internet Measurement Workshop (IMW 2002)*, pages 91–92, 2002.

[7] Dingde Jiang and Guangmin Hu. Large-scale IP traffic matrix estimation based on backpropagation neural network. In *Proceedings - The 1st International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2008*, 2008.

[8] Erol Gelenbe. Random Neural Networks with Negative and Positive Signals and Product Form Solution. *Neural Computation*, 1989.