User Guide

This is a simple user guide for the GMNS data format version of Big data-driven transportation computational graph framework (BTCG).

1. Introduction

This code is built on the GMNS data format version of BTCG (Wu et al., 2018). BTCG is a forward and backward propagation algorithmic framework on a layered computational graph which can achieve hierarchical travel demand estimation using multiple data source.

The specific relationship between the layers of HFN is shown in Fig1. HFN of Traffic Demand Flow Estimation (TDFE) model. The figure shows the input/output variables of each layer and the correlation calculation between layers.

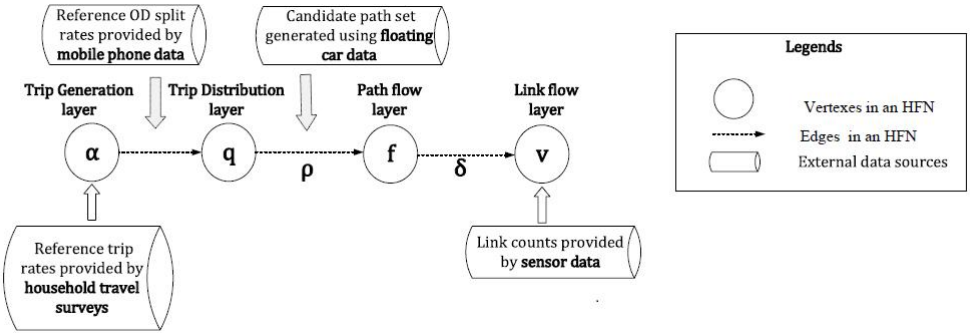


Fig. 1. HFN of TDFE model

BTCG is the implementation of Hierarchical Flow Network (HFN), by applying the back propagation (BP) algorithm, and by viewing each variable as a vertex and the edge between vertexes as calculation process between variables. To minimize the loss function, parameters like α, π and θ are updated during the training process. Finally, the traffic flow of each layer (e.g. α of ozone layer, γ of OD layer, v of link layer) are estimated jointly.

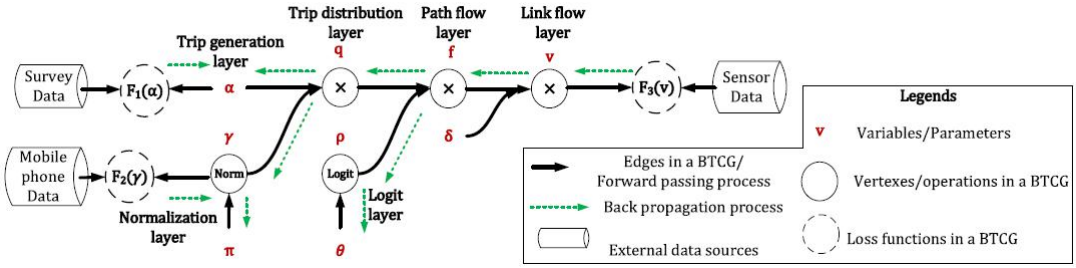


Fig. 2. Corresponding BTCG of HFN

2. Data flow

Input files	Output files
node.csv	output_ozone.csv
input_agent.csv	output_od.csv
agent_type.csv	output_path.csv
road_link.csv	output_link.csv

node.csv

This file node.csv may include the basic node information about the test network, such as

name, node_id, zone_id, node_type, ctrl_type, x_coord, y_coord, and geometry.

name	node_id	zone_id	node_type	ctrl_type	x_coord	y_coord	geometry
	1	0			1523373	1003235	POINT (1523373.000000 1003235.000000)
	2	0			1523873	1003225	POINT (1523873.000000 1003225.000000)
	3	0			1524263	1003205	POINT (1524263.000000 1003205.000000)
	4	0			1524224	1002265	POINT (1524224.000000 1002265.000000)
	5	0			1523854	1002765	POINT (1523854.000000 1002765.000000)
	6	0			1523354	1002745	POINT (1523354.000000 1002745.000000)

road_link.csv

Road_link.csv includes basic link-level information.

name	roadlink_i	from_nod	to_node_i	facility_ty	link_type	direction	length	lanes	free_spee	capacity	demand_t	geometry
	1	1	100002	HOT	1	1	277	1	25	1800		LINESTRING (1523373.000000 1003235.000000 ,1523448.678000 1002967.757000)
	1	1	101990	HOT	1	1	239	1	25	1800		LINESTRING (1523373.000000 1003235.000000 ,1523365.110000 1003474.169000)
	1	1	101997	HOT	1	1	293	1	25	1800		LINESTRING (1523373.000000 1003235.000000 ,1523666.177000 1003224.848000)
	1	1	101998	HOT	1	1	230	1	25	1800		LINESTRING (1523373.000000 1003235.000000 ,1523143.000000 1003235.000000)
	2	2	100004	HOT	1	1	264	1	25	1800		LINESTRING (1523873.000000 1003225.000000 ,1523854.344000 1002961.414000)
	2	2	101992	HOT	1	1	226	1	25	1800		LINESTRING (1523873.000000 1003225.000000 ,1523854.344000 1003450.648000)
	2	2	101996	HOT	1	1	277	1	25	1800		LINESTRING (1523873.000000 1003225.000000 ,1524150.707000 1003215.440000)
	2	2	101997	HOT	1	1	206	1	25	1800		LINESTRING (1523873.000000 1003225.000000 ,1523666.177000 1003224.848000)
	3	3	101995	HOT	1	1	170	1	25	1800		LINESTRING (1524263.000000 1003205.000000 ,1524432.957000 1003196.623000)

input_agent.csv

All the measurements are currently stored in input_agent.csv. Our code requires the household survey data (ozone data), the OD reference volume or the OD split rate (mobile phone data), the link count (sensor data), and the path information such as its node sequence.

agent_id	agent_type	from_zone_id	to_zone_id	from_node_id	to_node_id	od_flow	node_sequence	path_flow	time_peroid	observations
1	1	3	-1	3	-1	-1	-1	-1	1	70
2	1	12	-1	12	-1	-1	-1	-1	1	60
3	1	4	-1	4	-1	-1	-1	-1	1	90
4	1	6	-1	6	-1	-1	-1	-1	1	70
5	1	8	-1	8	-1	-1	-1	-1	1	30
6	1	9	-1	9	-1	-1	-1	-1	1	50
7	1	10	-1	10	-1	-1	-1	-1	1	60
8	1	11	-1	11	-1	-1	-1	-1	1	100
9	1	16	-1	16	-1	-1	-1	-1	1	90
10	1	2	-1	2	-1	-1	-1	-1	1	80
11	1	7	-1	7	-1	-1	-1	-1	1	50
12	1	18	-1	18	-1	-1	-1	-1	1	80

agent_type.csv

agent_type.csv is used to explain the agent type in input_agent.csv. Currently, it includes agent types :1. household survey; 2. OD reference volume; 3. path proportion; 4. link count, the attribute 1 means the number of observations, and attribute 2 means the number of time period. In our code, we consider one day of traffic volume, so the number of time periods is 1.

agent_type_id	agent_type	attribute1	attribute2
1	household survey	22	1
2	od reference volume	33	1
3	path proportion	126	1
4	link count	76	1

3. **Case study**

The Sioux Falls network is used as the test case, it has 22 zones, 24 nodes, 33 OD pairs, 76 links and 132 paths, the network structure is shown in Fig. 3. The GMNS format data is

provided in the sub folder of /SiouxFalls network.

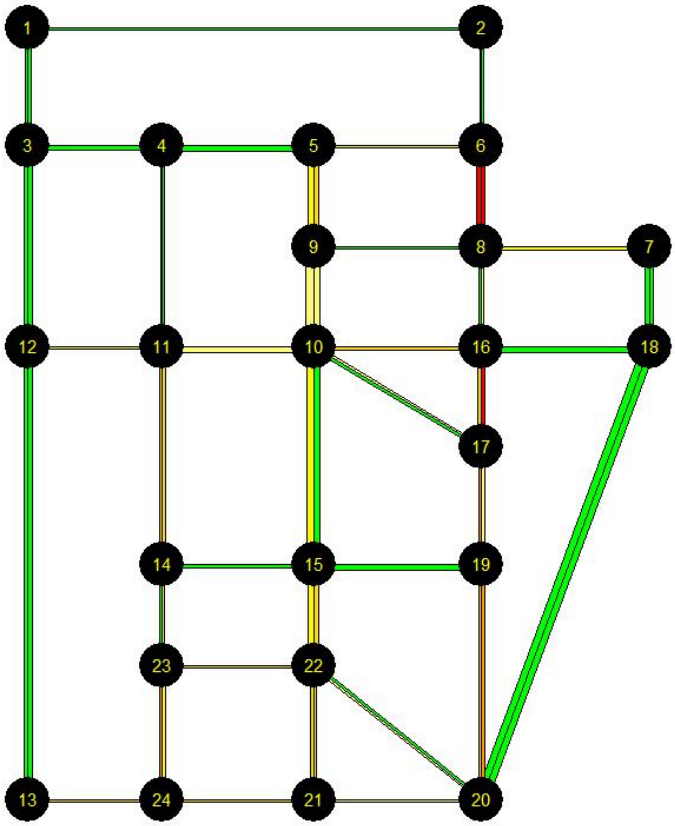
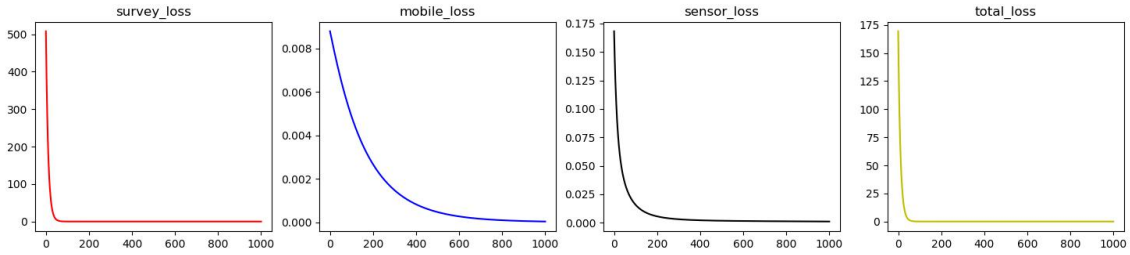


Fig. 3. Sioux Falls Network

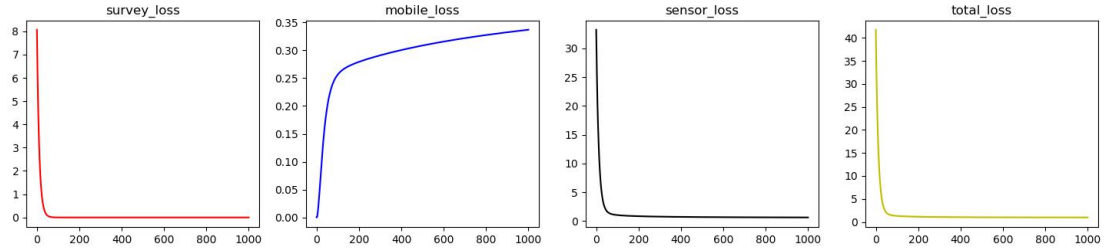
Training results (1000 epochs):

BTCGLite_GMNS.py (this code don't use logit model and class object, so it runs faster):



step 1000 :survey error= 6.983434152935154e-08
step 1000 :mobile error= 3.8845606447143905e-05
step 1000 :sensor error= 0.0010404423796513053
step 1000 :total error= 0.0010792072388590332

BTCG_GMNS.py Although the mobile loss increases to a certain degree,the total loss has been reduced, to demonstrate how the BTCG works.



step 1000 :survey error= 3.1824046e-09
step 1000 :mobile error= 0.3365797
step 1000 :sensor error= 0.6412415
step 1000 :total error= 0.9778478

References.

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transportation Research Part C: Emerging Technologies 96, 321-346.