HAMK University of Applied Sciences
and Ramboll Group

Project Report

Transportation modeling and Web content

František Kolovský
Pierre-Vincent Vrot

Erasmus Exchange from
University of West Bohemia
and IMERIR

November 23, 2015
version 0.1

Abstract

The Geographic Information Systems (GIS) allow, among others, to acquire, to treat, to organize and to present geographical data, to produce clear, accurate and intuitive maps, via an accessible web component from any browser.

The TraMap project, proposed by HAMK and supervised by Ramboll, is a simple and flexible tool that allows a user to view the transportation habits of information from open source geographic data and show informations in an application (web or mobile).

It in this context, of consultation tool and operating data, that our project takes place. Our role is to look for open data sources available on the Internet, to extract the information, to build an usual transport model and then develop a consultation multiplatform web application.

Keywords: GIS, Transport Modeling, Web, Algorithm, Open Sources

Acknowledgments

acknowledgments.tex

Contents

Introduction

In Finland, cycling, walking and public transport are increasingly used, it becomes necessary to bring the most useful information possible means in order to anticipate the movements.

In the current era, companies are always on the lookout for new products, particularly in the GIS domain, either issue new tools or technology. Each companies develop and market their traffic information tools. That is why our project is centered on the sharing of information and free access to sources and data.

Today, open source and open data sources of information are increasingly used and allow to evolve in cooperative community. Our main objective for this project is to recover the data and open source tools on which anyone could build or develop a data model of consultation on transportation habits of a city. Our research work will focus on the city of Hyvinkää and bicycle habits data.

The work is divided into three parts: the first is to determine the theoretical and transportation model data to be consumed; the second explains how an application can be developed for the retrieval of data; and the third and final section combines the previous parts in order to get a result viewed on different platforms.

1 Transportation Modeling

1.1 Transportation model

In upcoming section following expressions will be used:

T	transportation matrix (number of trips between zones)
C	travel cost matrix (between zones)
T_i, T_j	sum value in row/column in T
n	number of zones (size of matrix)

1.1.1 Trip destination

For determine transportation matrix T was used Gravity model:

$$T_{ij} = K_i K_j T_i T_j f(C_{ij})$$

where

$$T_i = \sum_{j=1}^{n} T_{ij}$$

$$T_j = \sum_{i=1}^{n} T_{ij}$$

$$K_i = \frac{1}{\sum_j K_j T_j f(C_{ij})}$$

$$K_j = \frac{1}{\sum_i K_i T_i f(C_{ij})}$$

where in our case

$$f(x) = x^{-2}$$

. T_i is number of trips outcoming from the zone (origin in the zone) i, T_j is number of trips incoming to the zone (destination in zone) j. So sometimes transportation matrix is called *OD Matrix*.

Now we must determine K_i and K_j. We used iterative proportional fitting. It is iterative solution. Fist we compute T^1 with $K_i, K_j = 1$. After we can use iteration equation for T:

$$T_{ij}^m = \frac{Z_i}{T_i^{m-1}} T_{ij}^{m-1}$$

$$T_{ij}^m = \frac{Z_j}{T_j^{m-1}} T_{ij}^{m-1}$$

where Z_i and Z_j are origin and destination trips (we know), m is iteration.

1.1.2 Count traffic

Now we know how many people travel from zone i to zone j, so we can find path from i to j and attributed this value into every edges in path.

In our solution we compute N paths for every pair of zones. Every path is based on another cost. Cost is based on length, time and vertical distance. Final cost is linear combination these partition cost.

$$c = \begin{pmatrix} k_t & k_l & k_h \end{pmatrix} \begin{pmatrix} t \\ l \\ h \end{pmatrix}$$

where c is cost, t is time, l is length and h is vertical distance. Number of trip is split evenly among all the paths.

1.2 Implementation

In upcoming section following expressions will be used:

n	number of nodes
m	number of edges
z	number of zones
p	number of path computed for one pair of zone

Transportation modelling described in previous section was implemented in Python programming language.

1.2.1 Static shortest path search

For determined C we use Dijkstra's algorithm (complexity $O(m + nlog(n))$). So final complexity is

$$z(m + nlog(n))$$

For traffic count we used also Dijkstra's algorithm. Final complexity for traffic count is

$$pz(m + nlog(n))$$

For Dijkstra's algorithm we used Python library iGraph. iGraph is written in C programming language so it is fast. For example one Dijkstra running 30 ms ($n = 8000$ $m = 18000$).

1.2.2 Data store

All data for transportation modelling are stored in relation database PostgreSQL with extension PostGIS. In database there are 4 main table:

roads	road links (edges)
nodes	nodes (vertexes)
zones	list of zones
traffic	
general_area_information	contains interested area geometry
od_pair	DB implementation for T matrix

More details about DB you can find in project documentation on GitHub.

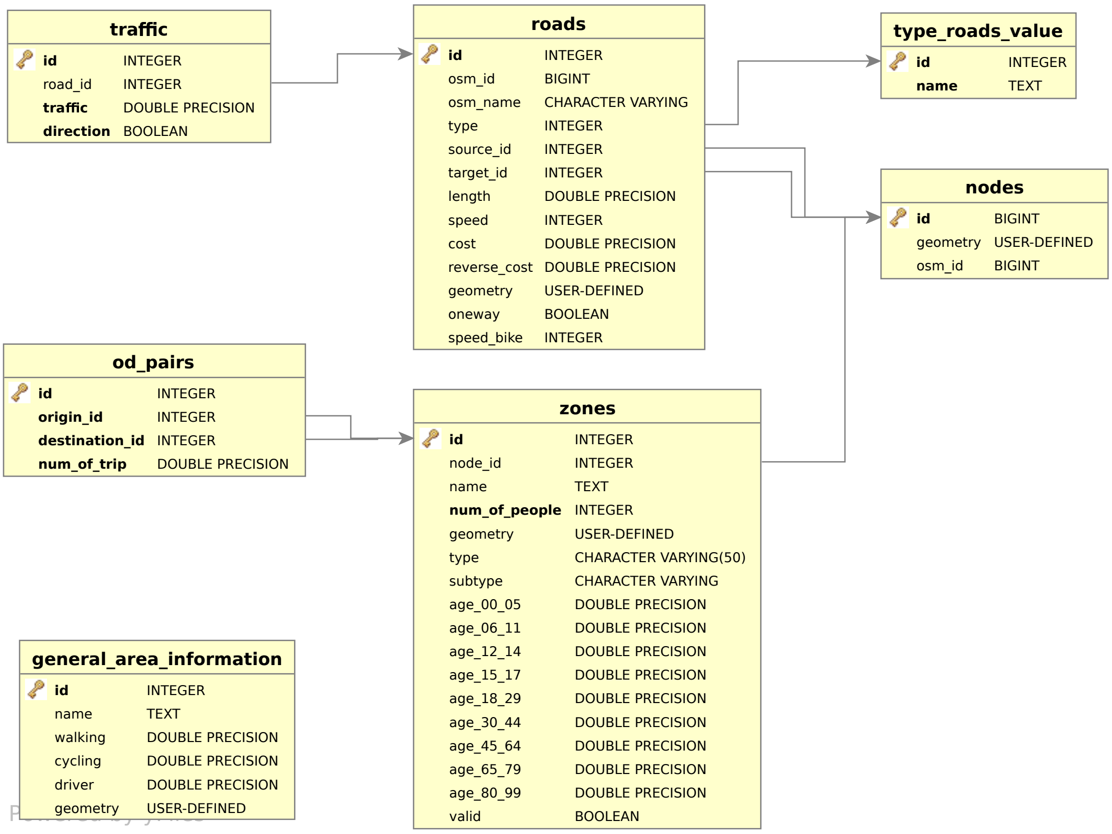

Figure 1.1: DB model

2 Web content

2.1 Prototype

s02-code.tex s03-practice.tex

3 Merge

s01-data.tex s02-app-model.tex

Conclusion

conclusion.tex

Bibliography

[REF] `http://resources.arcgis.com/fr/help/`
ArcGIS 10.3 Documentation. Environmental Systems Research Institute, Inc, 2015.

List of Figures

List of Tables