Complex Networks, Project (Bicing barcelona)

Andrés F. Lamilla

June 24, 2015

Contents

1 Introduction

The idea of this project is to analyse the bicing system from barcelona using complex networks to modelate the movement of the bikes between all the stations.

The bicing system has 420 stations sparse around barcelona, and each of them has a capacity of around 20 bikes.

This system has a web api that can be consulted from anyone and return a xml file with information about every station. Ex:

```
<bicing_stations>
 <updatetime><![CDATA[1432677551]]></updatetime>
 <station>
  <id>1</id>
  <type>BIKE</type>
  <lat>41.397952</lat>
  <long>2.180042</long>
  <street><![CDATA[Gran Via Corts Catalanes]]></street>
  <height>21</height>
  <streetNumber>760</streetNumber>
  <nearbyStationList>24, 369, 387, 426</nearbyStationList>
  <status>OPN</status>
  <slots>0</slots>
  <bikes>24</bikes>
 </station>
 <station>
  <id>3</id>
  <type>BIKE</type>
  <lat>41.393699</lat>
  <long>2.181137</long>
  <street><![CDATA[Ali Bei]]></street>
  <height>21</height>
  <streetNumber>44</streetNumber>
  <nearbyStationList>4, 6, 119, 419</nearbyStationList>
  <status>OPN</status>
  <slots>0</slots>
  <bikes>25</bikes>
 </station>
 <station>
  ...
```

The most important information here are the number of bikes and slots available in each station, the geographical position of the station and the time when this information was obtained. The status field says if the station is working or not.

With this data we can get a graph of the stations in that time.

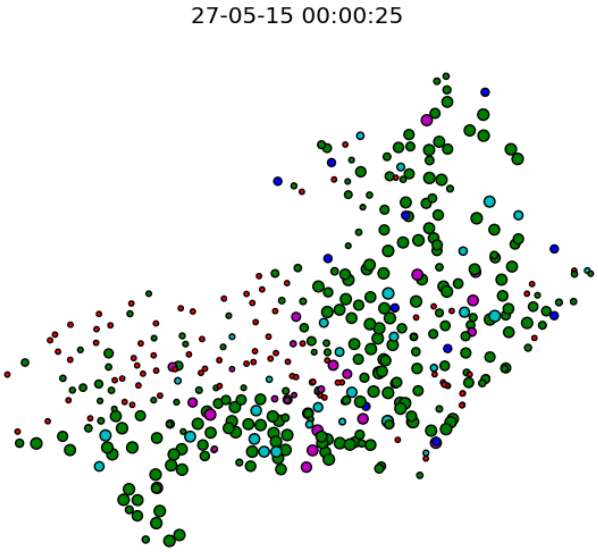

Figure 1: Graph of the bicing stations in BCN

Every circle on the graph indicate one station. The size of the circle gives an idea of the number of bikes in the station. A bigger circle show a high number of bikes in the station. A blue circle indicate a not working station, a green circle indicate a station with bikes and a red circle indicate a station without bikes. To get even more information from the graph, I decide to save the number of bikes in the previous time, and show if there was an increment on the number of bikes in the station or if the number of bikes decrease than the previous time. For this, a cyan circle indicate if the station has more bikes than before and a magenta circle indicate if the station has less bikes than before.

Until here we have a big picture of the state of the stations in a given moment, but we don't know how the bikes are moving in the city. To do this I had to give an estimation of how long it takes to go from one station to the rest of stations. And after that decide how are the posible movement of the bikes in the city given the changes of the number of bikes in the stations. I used the distancematrix api (https://maps.googleapis.com/maps/api/distancematrix/json) from google to get an idea of how long it takes to go from one point to another by walking. I had to scale this number by a factor of 2.13 to get an approximation for bikes. And also I include a confidence interval of 20% to get a range of all posible durations. I got this values with information of my own tracks in some stations. The google api has a limitation on the number of queries so I had to use the vincenty algorithm from geopy (https://pypi.python.org/pypi/geopy) as a fallback.

I plot this possible routes for all the new arrives in the last minute, the last five minutes and the last fifteen minutes.

To visualize all the resulting images I build a little web app using javascript.

2 Code

The main code was written in python. I used used networkx library for python to build the network.

The code was tested on a linux machine using ubuntu 14.04 and python 2.7. It's necessary to install networkx and matplotlib librarys for python. The requirements are in code/requirements.txt file and can be installed using pip install -r requirements.txt .

The raw data from bicing (xml files) is not included in the code because it is too big. You have to download it from http://static.lamillac.com/cn_proyect/public/data_day.tar.gz

I also have a data.json file with the xml processed information in

http://static.lamillac.com/cn_proyect/public/data.tar.gz .

This data files should be saved after decompress in the code directory.

I also write some vagrant (https://www.vagrantup.com/) files to build a virtual machine with all the requirements for this project. You just have to run "vagrant up" from the Vagrant directory to install the virtual machine and the run "vagrant ssh" to get a console from the virtual machine. The code in the virtual machine is located in /opt/code/.

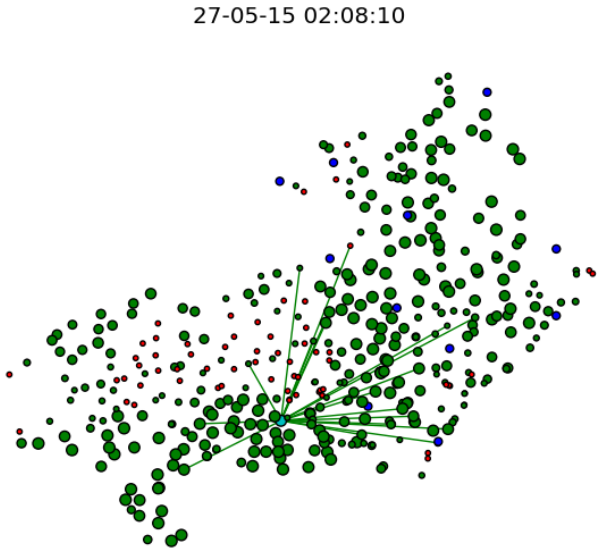

Figure 2: Graph of the arrives in the last minute

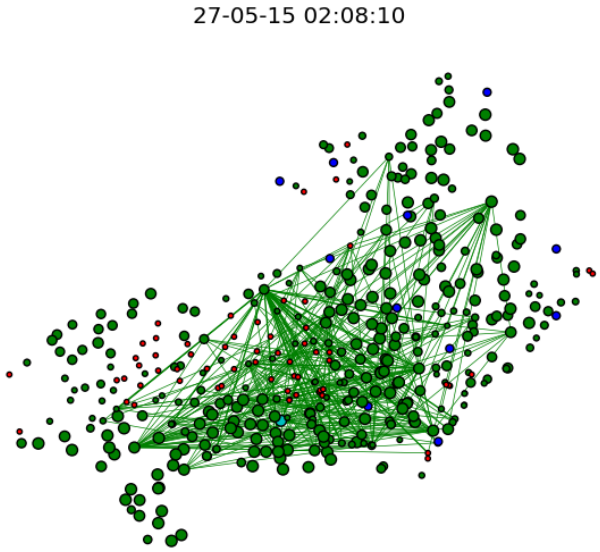

Figure 3: Graph of the arrives in the last five minute

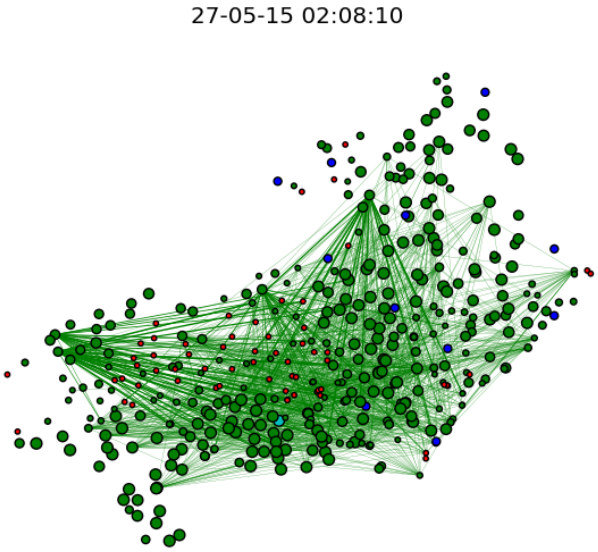

Figure 4: Graph of the arrives in the last fifteen minute

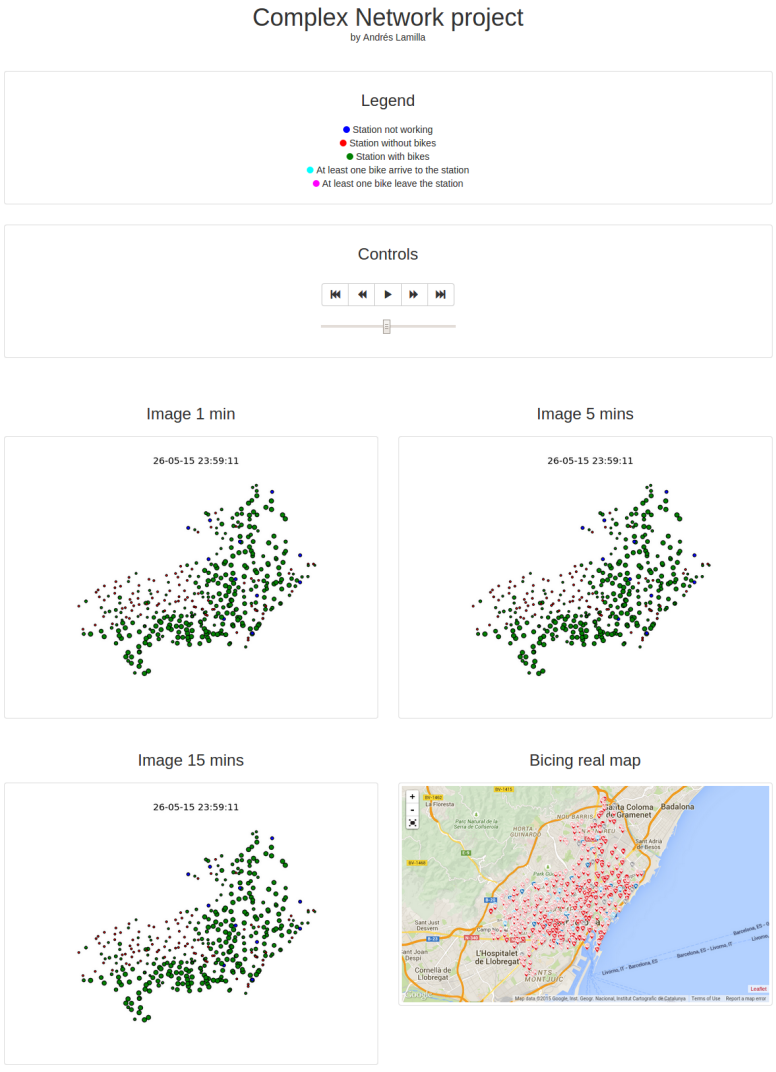

Figure 5: Project web app

If you just want to show the results, you have to download all the images results from http://static.lamillac.com/cn_proyect/public/images.tar.gz and save them in code/public. Then just open the index.html file from code/public in any modern browser.

3 Results

I only processed one day of the obtained data because it is very time consuming. But with this data we can see how the bikes are constantly moving between stations. I wanted also to get some communities with this information but I couldn't finish this task. I think that it is posible to get an idea of the more transited stations and their routes doing this.