Analysis of Pass Distribution Networks on Football Teams

André Diegues

July 2, 2016

DCC/FCUP

Contents

1 Introduction

What can we say to describe what is or what is not a network? A network is a set of vertices or nodes, with connections between them, called edges. Systems that take the form of networks are designated by graphs and almost everything can be represented in this form. Nowadays, networks fit to represent some of the most important things in the world, like social networks like *Facebook* or *Twitter*, biological networks such as neurological networks, information networks like the network of citations between academic papers and, lastly, technological networks which are all networks that are implemented by the human kind in the world such as water networks, transportation networks and even communication networks [4].

In this article we're going to represent through networks the pass distribution of football teams in the high-level of the sport looking specifically to the best European competition, the *UEFA Champions League*. We are hoping to find patterns in the networks that can reveal, for example, who is the most influential player of a team, how the absence of that player impacts the performance of his team and also if the team's strategy is the same with and without that player.

2 Building the Networks

In order to represent all the teams in the *UEFA Champions League* we had to get the pass distribution of these teams. Fortunately all the data from the season 2012/2013 till the present season was available on the *UEFA* official website at `http://www.uefa.org/mediaservices/presskits/uefachampionsleague/` in the total pass distribution .pdf files. We can see Figure 1, in the following page, as an example of these files.

UEFA CHAMPIONS LEAGUE

Passing Distribution

Semi-finals 1st leg - Wednesday 6 May 2015

Camp Nou - Barcelona

FC Barcelona 3 - 0 FC Bayern München

From		TP	To: 1 Marc-André ter Stegen	3 Gerard Piqué	4 Ivan Rakitić	5 Sergio Busquets	8 Andrés Iniesta	9 Luis Suárez	10 Lionel Messi	11 Neymar	14 Javier Mascherano	18 Jordi Alba	22 Daniel Alves	6 Xavi Hernández	12 Rafinha	15 Marc Bartra
ter Stegen	1	95'09"		3	-	2	1	1	1	-	5	11	2	-	-	-
Gerard Piqué	3	95'09"	5		5	3	2	2	1	-	2	2	4	-	-	-
Ivan Rakitić	4	81'58"	1	3		11	-	4	8	3	1	2	11	-	-	-
Sergio Busquets	5	95'09"	1	4	7		9	1	2	2	4	8	7	-	-	-
Andrés Iniesta	8	86'35"	-	1	4	6		2	4	7	4	10	3	-	-	-
Luis Suárez	9	95'09"	-	-	3	1	-		6	2	-	1	-	-	1	-
Lionel Messi	10	95'09"	-	-	9	3	3	6		6	-	2	2	-	-	-
Neymar	11	95'09"	-	-	2	1	7	4	4		1	7	1	-	-	-
Javier Mascherano	14	88'26"	7	4	2	4	5	1	1	-		4	3	-	-	-
Jordi Alba	18	95'09"	4	-	1	3	8	5	4	16	6		-	-	-	1
Daniel Alves	22	95'09"	-	4	14	2	3	1	12	1	2	1		-	-	-
Xavi Hernández	6	13'11"	-	-	-	-	-	-	-	-	-	-	-		-	-
Rafinha	12	8'34"	-	1	-	-	-	-	-	-	-	-	1	-		-
Marc Bartra	15	6'43"	-	-	-	-	-	-	-	-	-	-	-	-	-	
Total passes received:			18	20	47	36	38	27	43	37	25	48	34	0	1	1

From	Long PC	Long PA	Medium PC	Medium PA	Short PC	Short PA	Total PC	Total PA	Total %
ter Stegen	11	17	12	12	3	3	26	32	81%
Gerard Piqué	4	7	17	18	5	5	26	30	87%
Ivan Rakitić	2	3	24	29	18	20	44	52	85%
Sergio Busquets	6	6	24	26	15	19	45	51	88%
Andrés Iniesta	2	2	21	24	18	20	41	46	89%
Luis Suárez	0	1	9	13	5	9	14	23	61%
Lionel Messi	4	4	16	17	11	16	31	37	84%
Neymar	0	1	13	14	14	18	27	33	82%
Javier Mascherano	6	9	21	22	4	5	31	36	86%
Jordi Alba	3	4	29	34	16	20	48	58	83%
Daniel Alves	2	2	22	26	16	20	40	48	83%
Xavi Hernández	0	0	0	0	0	0	0	0	0%
Rafinha	0	0	2	2	0	0	2	2	100%
Marc Bartra	0	0	0	0	0	0	0	0	0%
Total	40	56	210	237	125	155	375	448	84%

From		TP	To: 1 Manuel Neuer	3 Xabi Alonso	5 Medhi Benatia	6 Thiago Alcántara	9 Robert Lewandowski	13 Rafinha	17 Jérôme Boateng	18 Juan Bernat	21 Philipp Lahm	25 Thomas Müller	31 Bastian Schweinsteiger	19 Mario Götze
Manuel Neuer	1	95'09"		4	5	1	1	7	4	4	2	1	2	-
Xabi Alonso	3	95'09"	3		5	17	1	10	7	5	11	3	13	1
Medhi Benatia	5	95'09"	9	8		3	2	8	7	-	6	3	1	1
Thiago Alcántara	6	95'09"	-	8	7		4	5	6	10	4	7	8	3
Robert Lewandowski	9	95'09"	-	2	-	1		1	-	4	3	1	2	-
Rafinha	13	95'09"	7	15	8	5	-		3	2	4	3	2	1
Jérôme Boateng	17	95'09"	7	3	5	1	3	2		5	4	-	5	-
Juan Bernat	18	95'09"	-	6	-	14	2	1	4		1	-	1	3
Philipp Lahm	21	95'09"	2	13	6	7	1	5	1	3		8	8	1
Thomas Müller	25	78'11"	-	1	1	-	8	2	-	1	6		5	-
Schweinsteiger	31	95'09"	-	6	2	9	5	3	2	6	10	4		3
Mario Götze	19	16'58"	-	2	1	2	-	1	1	3	-	-	1	
Total passes received:			28	68	40	60	27	45	35	43	51	30	48	13

From	Long PC	Long PA	Medium PC	Medium PA	Short PC	Short PA	Total PC	Total PA	Total %
Manuel Neuer	7	12	22	22	2	2	31	36	86%
Xabi Alonso	6	7	48	54	22	24	76	85	89%
Medhi Benatia	5	6	30	33	13	15	48	54	89%
Thiago Alcántara	4	5	41	43	17	18	62	66	94%
Robert Lewandowski	0	0	6	9	8	11	14	20	70%
Rafinha	2	4	36	38	12	14	50	56	89%
Jérôme Boateng	8	12	25	28	2	2	35	42	83%
Juan Bernat	0	0	17	18	15	20	32	38	84%
Philipp Lahm	1	1	33	36	21	24	55	61	90%
Thomas Müller	1	1	13	13	10	16	24	30	80%
Schweinsteiger	3	3	27	29	20	24	50	56	89%
Mario Götze	0	0	7	7	4	5	11	12	92%
Total	37	51	305	330	146	175	488	556	88%

TP: Time played PA: Passes attempted PC: Passes completed %: Passing success percentage

00:22:28CET
07 May 2015

UEFA Media Information

Figure 1: Example of a pass distribution file from the official *UEFA* website

2.1 Extracting the Data

Since the files are .pdf files, the data isn't readable, in that case we implemented a JAVA program that would download the file and right after the download would convert the .pdf file into a .txt file. This program would automatically invoke, to all the pass distribution files available in the website, the `wget` command following from the `pdftotext -raw` command (the `raw` option guarantees the integrity of the data).

2.2 Collecting the Data into Data Structures

Like in the section 2.1 we coded everything in JAVA. It is a object-oriented high level programming language with a very good API, a lot of documentation and community support, a very satisfying performance and exception handling that can provide flexibility when it comes to be creative. Creating a XML or SQL database wouldn't be necessary because we are not looking to be doing queries and JAVA has the particularity of free the allocated memory when the program terminates, the objective of this project is to study the pass distribution networks and all the queries can be easily done in JAVA if necessary.

Analysing the converted .txt files, the data was all spread out and not organized at all. Due to this fact we decided to implement a JAVA program to parse the data in order to organize and, most importantly, to make it reliable and consistent. After that, the primary focus was to create data structures that could represent all the data found in one file.

Given some consideration we decided that we wanted each team to have a graph for each different season. In order to do that we implemented the following data structures:

1. Match
   (a) MatchScore
   (b) MatchDate
2. Player
3. Team
4. Season
5. Pass

2.3 Converting the Data into Networks

Now that the data is collected we need a tool to reproduce the data into networks. In the following section we'll introduce you to Gephi: An Open Source Software for Exploring and Manipulating Networks [1].

2.3.1 Gephi

Gephi is a powerful tool to visualize and analyse networks. It has a friendly and easy-to-use framework and the task of importing data is as easy as it can be. Anyone can build networks from spreadsheets or another file extensions, such as GraphML, GML, CSV and so on [2]. The Figure 2 is an example of the capabilities of Gephi with respect to network visualization and analysis.

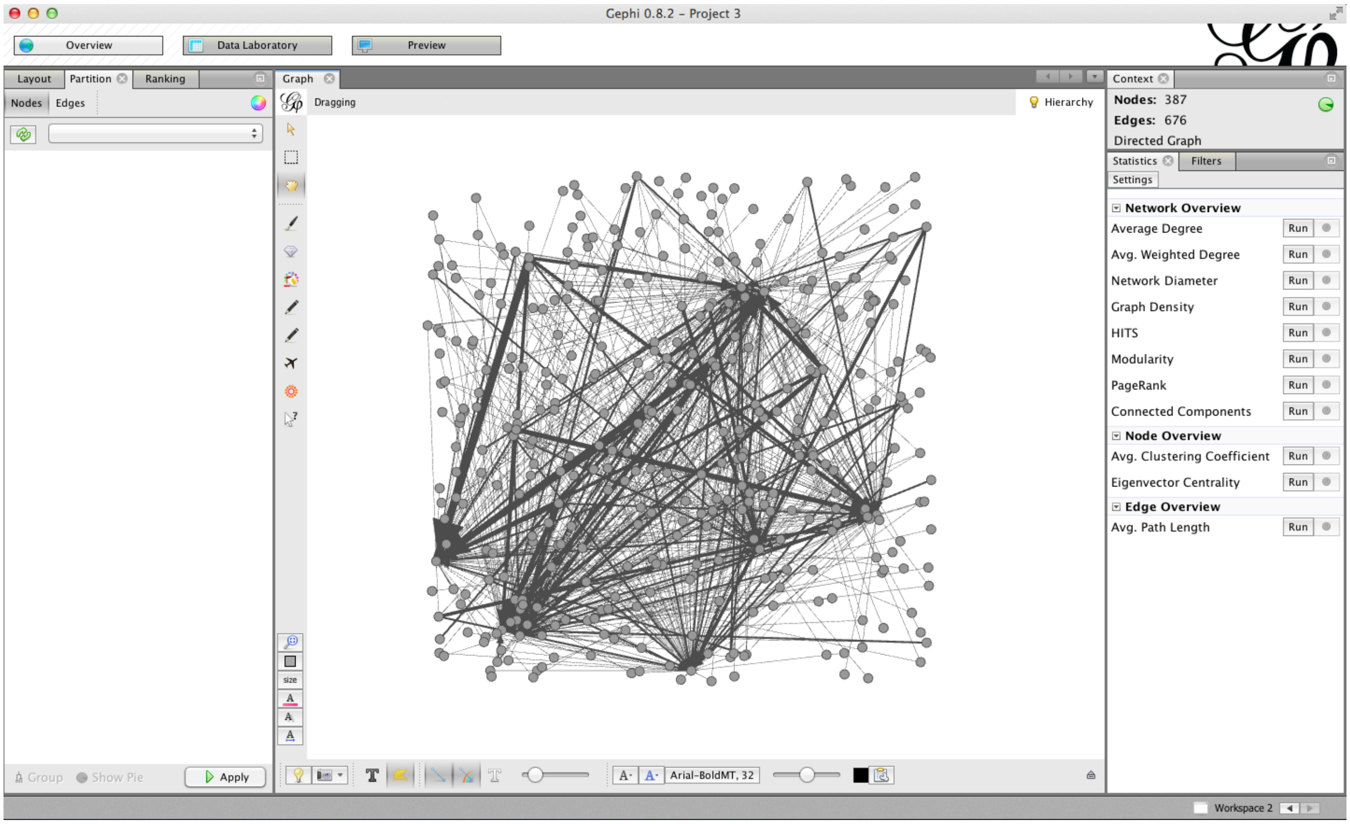

Figure 2: Example of a network built in Gephi

2.3.2 Parsing the Data

Since we're using Gephi to analyse our pass distribution networks, we needed to rearrange the data so it would be easier to import it to Gephi in the future. In this way, we would not waste time importing data to the platform. We've decided to parse the data in two formats:

1. Spreadsheet Notation

   The Spreadsheet Notation is very easy to read and understand the data. For this project, any spreadsheet presented like Tables 1 and 2 can be easily imported into Gephi, we need only to specify which table represents the nodes and which table represents the edges and after that the network is built nice and easy.

Id	Name
20	Sergio Romero
7	Memphis Depay
8	Juan Mata
...	...

Table 1: Example of nodes

Source	Target	Weight
20	10	3
20	11	1
20	12	8
...	...	...

Table 2: Example of edges

2. GML Notation

   GML, Graph Modelling Language, consists of a hierarchical key-value lists and it features portability, simple syntax, extensibility and flexibility [3]. As we can see in Figure 3, firstly we must declare the nodes of the network before the edges. By doing this, we guarantee that the hierarchical key-value lists condition is preserved, avoiding information to be lost in the process of construction of the network.

```
graph [
	directed 1
	comment "Sample comment"
	node [
		id "A"
	]
	node [
		id "B"
	]
	node [
		id "C"
	]
	edge [
		source "A"
		target "B"
		label "Edge from node A to node B"
		weight "0.1"
	]
	edge [
		source "B"
		target "C"
		label "Edge from node B to node C"
		weight "0.5"
	]
	edge [
		source "C"
		target "A"
		label "Edge from node C to node A"
		weight "0.5"
	]
]
```

Figure 3: Example of a GML network

3 Results

References

[1] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. 2009.

[2] K. Cherven. *Network Graph Analysis and Visualization with Gephi*. Community experience distilled. Packt Publishing, 2013.

[3] Michael Himsolt. Gml: A portable graph file format. *Html page under http://www. fmi. uni-passau. de/graphlet/gml/gml-tr. html, Universität Passau*, 1997.

[4] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.