VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam
The Netherlands

Online TV Buzz

An analysis of the interaction between Social Networks and TV audience

BMI MasterThesis

Author:
Valentina
Maccatrozzo

Supervisors:
Dr. Stefan Schlobach
Dr. Mauro Barbieri
Second reader:
Dr. Sandjai Bhulai

August 15, 2011

Abstract

This thesis aims at analyzing the possible interactions between the popularity of a TV program episode and its features.We propose an innovative approach in order to grasp these possible connections. The idea is that of building a network, which changes over time, where the nodes represents the guests of the episode. The edges of the network are due to the co-participation of the guests to the same episode. We aim at showing that the popularity of an episode is connected to the popularity of its guests. The popularity of the guests is defined on the basis of centrality measures we extract from the network, while the popularity of a TV program episode is defined on the number of the tweets posted while the episode is on air, that contain the title of the program. We show that there are some connections between the popularity of the guests and that of an episodes as we defined them, however these are more marked in TV programs types for which guests are really fundamental and that have titles which are more recognizable in the Twitter cloud. Furthermore, we propose a new ontology to represent Electronic Program Guide (EPG) in a novel way.

Preface

This thesis is part of acquiring the Master degree in Business Mathematics and Informatics at the VU University of Amsterdam and is the final report of an internship undertaken at Philips Research in Eindhoven.
The subject of this research is the analysis some possible networks properties that influence the construction of a good predictive model for the popularity of a TV show. In particular, we aim at analyzing the possible existing interactions between the popularity of a TV program episode and its guests. This aim is reached by mean of a social network built with the guests of the episodes, where the edges represent a co-participation to the same episode. We will investigate the characteristics of such network, namely the centrality measures of the nodes interpreted as popularity measures of guests, in order to grasp the possible connections between the popularity of an episode and that of its guests. We will show that such connections exist, if some prerequisites are satisfied.
I would like to thank all the people who allow the success of this Master Project. In particular I would like to thank all my supervisors, starting from the external ones, Mauro Barbieri, Jan Korst and Ramon Clout, and the internal one, Stefan Schlobach, who supported me the most. Finally, I would like to thank my family, and in particular my partner for all the support he always gives me.

Valentina Maccatrozzo
Amsterdam, 2011

Contents

Chapter 1

Introduction

1.1 Social TV: a worldwide living room

> *The ability to create content that will enable people to interface with each other, to connect, to recommend, to share and experience over television, is going to change the landscape of the industry* [22].
>
> *Ynon Kreiz, CEO of the Endemol Group*

Social TV is a broad term to indicate interactions about TV: talking about a TV program while watching it or talking about related content. In a certain way television has always been a social activity. Since television first appeared in our living room, it became a way of spending time together. It was also a mean to unify people and families. Since then, people have started to talk about TV, especially if there was something interesting to comment on.

Until recently, the television was considered by many as a one-way medium where the content comes through but nothing goes back. The TV generally goes off and the computer goes on when people want to connect and communicate with others. But now, some of the tools that allow people to build communities and socialize on the Internet are making their way to the living room. Social networks allow quick interaction which imitate a real chat. More and more connections will lead to a worldwide interaction about TV through the Web.

For example, *Twitter* [39] offers a social networking and microblogging service, enabling its users to send and read messages called "tweets". It is a

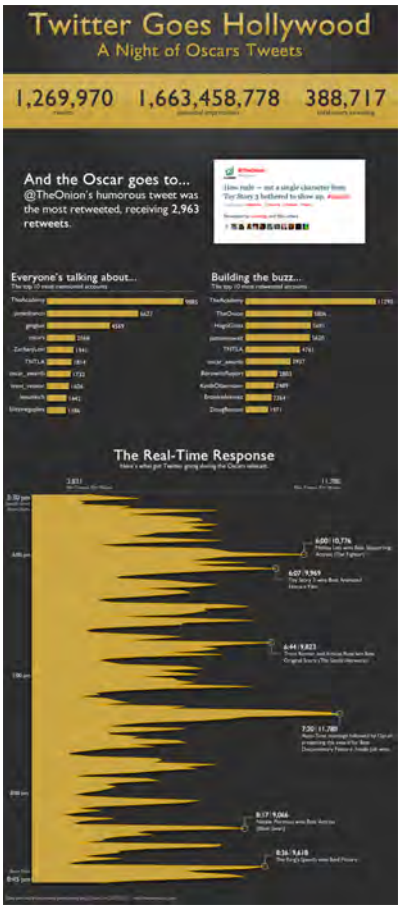

Figure 1.1: Mass Relevance and *TweetReach* [6], a Twitter analytics service with commercial access to the Twitter API, have teamed up to make a data map of the mass conversation about the Oscar Award Ceremony

perfect tool to facilitate interaction on TV program episodes in real time. If you are looking at a TV program and you are physically alone, being on Twitter allows you to be together with your friends digitally. And everything you otherwise would have said you can now post. Let us consider the tweets that are posted during this year's Academy Awards Ceremony as presented in Figure 1.1. Over 20 Oscar-related terms like 'Oscars', '#Oscars', 'Academy' (no specific names of celebrities or movies) were tracked during the show's live airing. The final numbers are: 1.269.970 tweets, 1.663.458.778 potential impressions[1], and 388.717 users tweeting [3]. Probably, on TV alone, the

[1]A potential impression is an indication of how many times a keyword is mentioned in the posts and the potential is based on the assumption that every follower actually views

Oscars would not have created the same "buzz". But on social networks the conversation can have been drawn on many different aspects of the show: fashion, politics, general gossip. And all this allows to improve the interest around the show.

So far we have spoken about an important show, which is always viewed by an huge amount of people. But what about every-day TV shows? Does Twitter in some way also help these programs in augmenting their audience? TV Genius [38], an application for TV recommendation and content discovery, has tracked tweets for about 90 UK unique shows over one week. Their results seem to confirm that people tweet a lot also about every-day shows. (see Table 1.1). Many companies are following on the Social TV trend. For

Day	Tweets	Top show
March, 20th 2011	10512	The Only Way is Essex, Duran Duran: One Night Only
March, 19th 2011	4673	Take Me Out
March, 18th 2011	6412	Comic Relief
March, 17th 2011	4046	Eastenders and Question Time
March, 16th 2011	9977	Masterchef and Waterloo Road
March, 15th 2011	10637	Eastenders
March, 14th 2011	12497	Coronation Street

Table 1.1: Most tweeted TV shows in the UK

instance, the BBC is including hashtags[2] on the screen at the start of a show, with some success. When *Question Time*[3] airs, it tends to top the tweets-per-hour rankings, with a variety of comments [12].

The promotion of the use of Twitter, and in general Social Networks, to call back people in front of the TV is now a trend followed by more and more TV producers. And the tracking of posts related to a specific TV programs is becoming an important measure to take into consideration when evaluating the success of a singular episode. Our contribution will focus mainly on the latter consideration, since we will provide a more structured approach in evaluating the impact of interacting social networks on the audience of TV programs.

the mentions of the keyword. This is an output of TweetReach.

[2]The # symbol, called a hashtag, is used to mark keywords or topics in a Tweet. It was created organically by Twitter users as a way to categorize messages.

[3]Question Time is a topical debate BBC television program in the United Kingdom, based on Any Questions?

1.2 The project

This study is part of a bigger project at Philips Research in Eindhoven: the Marmalade project. This project aims at developing algorithms and prototypes for recommendations and targeted advertising, predominantly in the TV and Internet video domain.

1.2.1 The dataset

The dataset on which we applied our experiments, is extracted from by Electronic Program Guide (in short EPG) in XML [46] format. EPG provide users of television, radio, and other media with menus displaying episode programming or scheduling information for current and upcoming programming. In particular our guide contains information about the time of the episodes, the duration, the title, the description, the type, the genres, the people involved, and some additional information about the episode (channel, if the episode is live or not). EPG are about German TV programs.

In total the dataset consists of about 300.000 TV programs, from which we extracted non-fictional programs with guests. We decided to concentrate our analysis on non-fictional TV programs. This decision depends on the fact that we want to analyze the influence of different guests in the TV shows over time. Fictional TV programs do not really help in this task: either they have always the same actors, in case of TV series, or they even do not have human actors, in case of cartoons. Our final dataset consists of 26.690 TV program episodes.

1.3 Goal and motivation

The main goal of this thesis is to analyze some possible social networks properties that influence the construction of a good predictive model for the popularity of a TV show. In particular, we aim at analyzing the possible existing interactions between the popularity of a TV program episode and its guests. In order to reach such a goal, we build two kinds of network, namely:

1. an EPG network. Nodes in this network are represented by TV program episodes, genres and guests. The edges are due to the episode genres and guests;

2. a social network. Nodes in this network are guests linked because they co-occurred in the same episode.

The use of Semantic Web Technologies (*i.e.* RDF [45], SPARQL [13], OWL [44]) facilitated the building task.

The advantage of using these networks is multifold.

First, they allow an innovative representation of the EPG, which underlines novel connections between TV program episodes. This could be particularly interesting, for instance, for a recommender system, which can take advantage of the graph properties to make new and interesting recommendations for eventual users of such a system.

Besides, the novelty of our approach resides in considering every singular episode as a stand-alone event in order to get a point of view similar to that of the viewer. To reach this objective, we decided to represent the network with an ontology especially developed for describing events, the Simple Event Model (SEM) [43]. The choice of SEM is due to the following reasons. First, simplicity, which is not meant as oversimplification, but as low commitment. This means that the model gives large degrees of freedom when describing events, without running the risk to loose internal consistency. Nevertheless, SEM contains all the elements necessary to properly describe events, and this is exactly what we require from it to accomplish. We individuate two layers of events: the first is the event related to the TV program itself (with its participants, and other details) and, second, the broadcasting event, with the consequent "buzz". We will mainly use the model to describe the first layer, but we aim at predicting the influence of this to the second layer evolution. Finally, SEM is also quite easy to extend, and so, we are easily allowed to enrich it in order to definitely fit our needs. This is also the reason why we do not adopt already existing ontologies for describing TV programs (for instance, see [9]): we do not focus on the mere technical aspects of TV broadcasting. We focus on the events related to it.

Finally, this approach allows to study the effect of changes over the time of the guests social network on the popularity of a show.

As popularity measure we decided to use the number of tweets that contains the name of the program while it is on air, as the authors of [7] did. This decision, partially justified in section 1.1, is also due to the fact that other popularity measures, such as the audience share, were not at our disposal. Tweets counts represent a good approximation of such a measure: they allow to measure all the "buzz" the episode produced. It is unlikely that people

who do not like the program, tweet about it while on air, so even without more specific analysis to distinguish between positive and negative tweets, as in [29], this can be consider a fair approximation of a popularity measure.

A strong predictive model for the popularity of a TV program episode can be useful to support different tasks:

- for the program producer: this can indicate on which guests the program should have or which kind of programs should be designed;
- for the program producer: this can indicate on which kind and at which price to sell space for advertising;
- for a recommender system: this can help in improving the recommendation, as we previously underlined.

1.4 Research focus

In general, our work aims at investigating whether the use of network information is useful for predicting popularity. Our research hypothesis is that there should be an interaction between the popularity of the guests, defined through centrality measures of the nodes in the networks, in an episode and that of the episode itself. In order to answer this question, we performed the experiment described in Section 3.4.1. The results, however, show a very poor influence of the popularity of guests on the popularity of the episodes.

A second research hypothesis is that there should be in our dataset a program type for which guests are really fundamental for the success of the episode. In order to answer this question, we performed the experiment described in Section 3.4.2. The results show that, actually, there is a specific program type for which our hypothesis holds, namely the program type *event*.

Then the research question we try to answer is whether there is a combination of features which gives the best performance in the classification task for the program type *event*. This question is answered by the experiment described in the Subsection of Section 3.4.2. The results show that there is a combination of features which improve the classification of 5.97% compared with the baseline.

And finally we want to investigate whether using a different popularity measure for the guests the popularity prediction is improved. In order to answer such question, we performed the experiment described in Section 3.4.3. The

results show that, in general, this popularity measure does not improve the prediction. However for a specific program type, namely *sport*, this measure improves the prediction of 2,56% with respect to the baseline.

Furthermore, we want to find what are the practical benefits of using Semantic Web technologies for the purpose of our practical goal. Answers to this question can be found in Section 2.2.

1.5 Related Work

Popularity prediction is a widely employed task within the social media environment. Nevertheless, this is one of the first scientific applications of the task to TV programs. However, some commercial tools already exist (e.g. see [38]).

Other applications of the popularity prediction task regards online content (for instance, see [18] and [32]). In particular, in [32] Jamali and Rangwala defined a co-participation network between users to analyze their behavioral characteristics and used the comment data available from Digg [11] to predict the popularity of online content. In [17] Szabo and Huberman present a method for predicting the long term popularity of online content using data from Digg and YouTube [51]. Another interesting paper is [27], where Satry, Yoneki and Crowcroft used information from social networks to predict the geographical access of online content.

A paper closer to our work, is [7], where Sitaram and Huberman demonstrated how social media content can be used to predict real world outcomes. In particular they used the chatter from Twitter to forecast box-office revenues for movies. They used the number of tweets containing keywords present in the title of a movie, and found out a strong correlation between these numbers and revenues at the box offices. Besides, they also apply sentiment analysis to improve the prediction.

SEM is a generic model for describing events, which is employed in a variety of heterogeneous fields, which include, for instance the naval (see [41]) or the historic domain (see [43]). SEM is mapped to concepts from other event ontologies such as the Event Model [50] (Queen Mary University), LODE [30] an ontology for linking descriptions of events, and the event model F [2].

The use of ontologies in the context of TV programs and EPG has already been explored also by other authors, but with different goals. In [49] Mizoguchi *et al.* use ontologies to calculate semantic similarities between TV

programs. In [19] Kim and Kang developed a personalized target advertisement system in iTV using ontology-based semantic relations among iTV contents for each viewer and stereotype. Another interesting work is [8], where Lui *et al.* developed an agent-based content management system for digital TV services with the help of ontologies.

1.6 Outline

Chapter 2 introduces the concept of network and explain how we built our networks and which kind of properties we extracted. Chapter 3 explains our experiments in details. Chapter 4 illustrates the results and Chapter 5 reports our conclusions and suggestions for further research.

Chapter 2

The Networks

In this Chapter we explain the concept of network, defying its properties and characteristics. We present the format and the ontology we choose to represent the networks. And, finally we characterize our networks and their properties.

2.1 What is a network?

Network is synonym of graph in the mathematical sense. A graph is a particular mean to represent objects connected each others because of some relation. Usually the interconnected objects are called vertices (or nodes), while the links are called edges (or arcs). More formally, a graph is an ordered pair $G = (V, E)$ comprising a set V of vertices together with a set E of edges, which are two-element subsets of V, *i.e.* an edge is related with two vertices, and the relation is represented as unordered pair of the vertices with respect to the particular edge [48]. Edges can also have a direction.

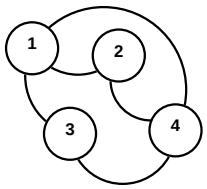

Figure 2.1: A simple graph with 4 vertices and 5 edges

If this is the case we define a graph as asymmetric (otherwise it is called symmetric). For instance, if the vertices represent scientific authors. There

will be an undirected edge between co-authors of the same article, while a directed edge will connect the authors and referenced ones.

2.2 Semantic Web Technologies

The work that we present in this thesis relies on Semantic Web Technologies. This section will introduce these technologies and will present the advantages and disadvantages of this choice.

Before starting, we should introduce the concept of Semantic Web. The Semantic Web is an initiative of the World-Wide Web Consortium (W3C). This is inspired by the vision of its founder, Tim Berners-Lee, of a more flexible, integrated, automatic and self-adapting Web, providing a richer and more interactive experience for users. The bigger vision found expression in an article written by Tim Berners-Lee, Jim Hendler and Ora Lassila in Scientific American in May 2001 [35]. In the words of the paper:

> *The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.*

The idea of Semantic Web is therefore to build a Web of data, partially relying on the information contained in the "traditional Web" which, by means of particular technology, makes this information easy to access and use by software agents. The technology that are proposed to satisfy this purpose are the following:

- XML
- XML Schema
- RDF
- RDF Schema
- OWL
- SPARQL

The first two technologies, namely XML and XML Schema represent the technological overlap with the Web. The first allows to produce documents that, although human readable, are strictly structured. The second is a language developed to allow to define the structure of these documents. These languages define the syntax that we will adopt to represent our data.

RDF, RDF Schema and OWL represent the semantic counterpart of XML and XML Schema. Whereas XML and XML Schema define the syntax of our documents, RDF, RDF Schema and OWL define the meaning (semantics) of their content. RDF is basically the language with which our data are represented. The syntax used to record these data is then defined by the XML serialization of RDF.

RDF Schema and OWL, instead, define the valid meaningful structure that we can adopt to represent our information. The basic component of RDF is the triple, a three element statement. These languages are used to build the so-called ontologies, which are documents that represent the logical relations and constraints that have to exist among our data. Equivalently to what XML Schema define for the syntax, ontologies define the boundaries for the meaning of our RDF statements.

Finally, SPARQL is a query language that, equivalently to SQL, allows to retrieve data of our interest from a repository of RDF statements.

As for any technology, Semantic Web ones show particular advantages and disadvantages, that we will introduce now. The flexibility and the lightness of the technology are their major advantages. From these two characteristics, follows that the above listed technologies are:

- easy to use: all these techniques are very straightforward;
- easy to query: once the RDF file is made, SPARQL allows to query it in all possible way one can think of;
- easy to remember: all the relations need to be defined only once;
- easy to expand: using RDF files and ontologies, the integration of information from other sources is allowed.

However the flexibility of the framework can also be seen as partial cause of one of its possible disadvantages. The flexibility is reached also by using a composed framework, and the utilization of this framework in situations that do not require scalability, interoperability, openness, etc. can be seen as an unnecessary overhead.

Representing a network, for instance, is possible within almost any programming language, by using adequate structures. Especially when a particular network analysis is not necessary, the need to represent the network itself by means of an ontology (which will probably need to be created or at least extended), and the need for semantic Web libraries or reasoners in order to properly handle it, could be seen as an unnecessary complication, both in terms of software requirements and of development. However, given that our requisites include the need to deeply analyze the network, the possibility to have a static representation of it (always accessible and not always at runtime) together with the previously introduced advantages, make us choose these technologies.

2.3 Network set up

In order to build both the networks, we need to extract the information contained in the EPG. What we need is: guests, genres, program type, start time, duration.

The most time consuming task was to extract all the guests names. The XML files contain a tag *<mprsn>*, which contains the names of all people involved in the program, *e.g.* anchormen, guests, actors, camera men and so on. But, names of guests could also appear in the description of the TV program (*<losyn>*, *<shsyn>*), which we took into consideration as well. First we added to an updated list of celebrities all the names that appeared in the tags. We ended up with a list of about 14000 names. Then the list had to be checked by hand to remove duplicates, since German names could be written in different ways. Finally, with the help of the *Lingpipe* [4] library, in particular using an exact dictionary-based chunking [5]. With this approach, we extracted the names from the descriptions making an exact match between names in our list and in the text. We did not make any distinction between the different rules of the people in the programs. In the remainder part of this thesis, we will use the word "guests" to indicate any kind of person that takes part in the TV program episode.

As infrastructure for the networks, we decided to use the Resource Definition Framework (RDF), exploiting the already mentioned SEM ontology, which we extended for our purposes. These aspects are explained in the following sections.

2.3.1 *SEM* ontology and its extension

The use of an ontology allows to be more specific about what the nodes and the edges in the network exactly represent. Properties of the nodes and relations between them are well defined, without misunderstandings. Ontologies help to avoid ambiguities and misunderstandings between terms used in different datasets. Furthermore, ontologies allow to add knowledge to data, in such a way that machines can understand it as humans do. In other words, ontologies allow to capture the semantics of the data. For instance, consider the health field, in particular datasets about disease concepts. An ontology can provide a structure of *inheritable, environmental and infectious origins of human disease to facilitate the connection of genetic data, clinical data, and symptoms through the lens of human disease*. The objective is to create a useful tool *for coupling disease concepts in model organisms to human disease concepts* [40]. Such an ontology can help in creating a unified and integrated dataset of biomedical data that is associated in human diseases which will help doctors to detect diseases and to collect data about it in a more structured way

In the EPG context, the use of an ontology can help to build a very interactive EPG, adding links to other sources, *e.g.* Wikipedia pages of people in the program or about the argument it is going to present, and so on. See, for example, Figure 2.4. However, we do not address this extension in this work.

In our case, we were looking for an ontology that could allow us to represent the TV program episodes in an efficient way, especially such that the different episodes could link with all the information that we have at our disposal. We decided to use SEM (*Simple Event Model*) [43], [28], [42], a schema for the semantic representation of events, identifying a TV program episode as an event. The ontology does not deal with the way data about events is stored, but only with the events themselves. The ontology focuses on modeling the most common facets of events: who, what, where, and when. These are represented respectively by the SEM core classes *sem:Actor*, *sem:Place*, *sem:Object* and *sem:Time*. Apparently, this ontology satisfies all our requirements but two, which were suitably covered through a small extension. Indeed, we added only two new classes: *epg:Title*, *epg:Genre*, that is, the title and the genre of a specific TV program, as defined by its EPG information.

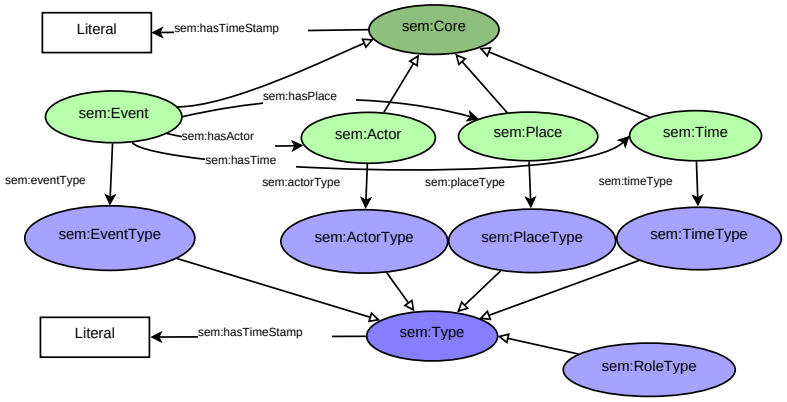

Figure 2.2: The classes of the Simple Event Model

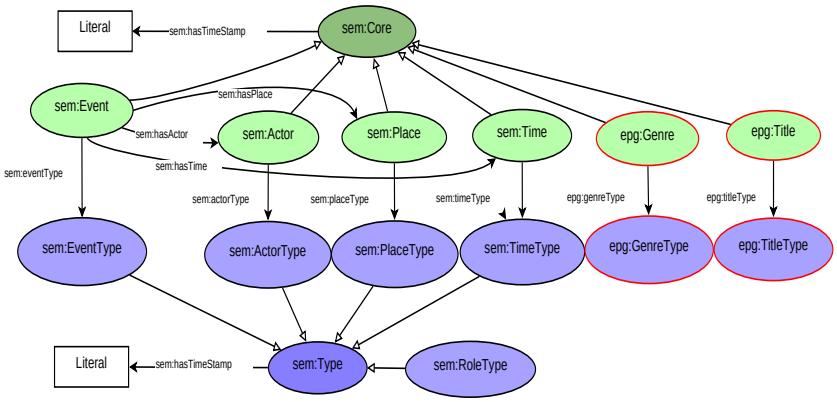

Figure 2.3: The classes of the extended Simple Event Model. New classes have red border.

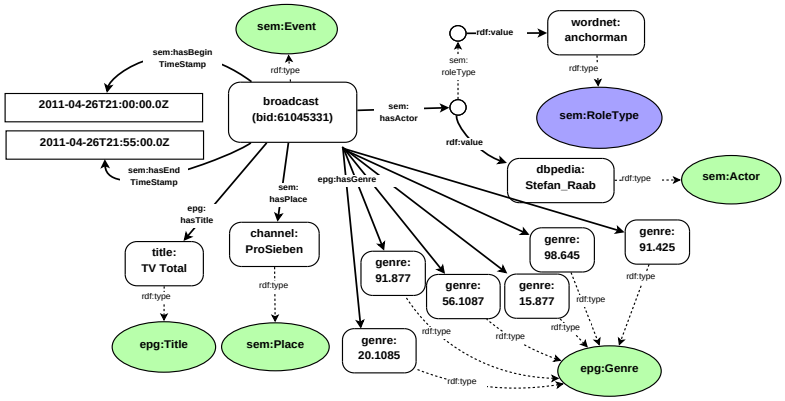

Figure 2.4: The *SEM* representation of a program.

2.3.2 *RDF* and *SPARQL*

We decide to represent our network using of the Resource Definition Framework (in short, RDF) language. RDF is a W3C[1] recommendation. RDF is a standard model for data interchange and meta-data representation on the Web, and beside that, the main reason for this choice resides on the fact that RDF is aimed at representing graphs, that is, networks. Of course there are many other ways to represent graphs, but RDF gives many advantages. First, RDF allows to keep track of the relations defined over time, which other representations do not allow to. Furthermore, RDF, as we show later, is very easy to query, and this is a great potential, especially if analyses from different points of view are required, as in this work. Indeed the guests network was extracted directly from the RDF representing the EPG.

RDF is defined in terms of "triples" (Subject, Predicate, Object), that can be seen as a sequence of a node, and edge and another node of a directed graph. Elements of these triples can be either literals (that is, strings), or URIs (that is a references to objects). Therefore, the language extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link. See Figure 2.5 for an example of the RDF representation of our data.

RDF can be represented through XML files (RDF/XML), Notation 3 (N3), a notation introduced by W3C, with Turtle or N-Triples. N3, based on tabular notation is the superset of Triple, which is the superset of N-Triples. We used the RDF/XML syntax.

With this serialization nodes (subject and objects) and predicates are represented in XML terms (see Figure 2.5. RDF URI's are represented as XML QNames. These are defined in the first parts where XML entities and RDF local names are defined. In this example, nodes are represented as RDF URI's and RDF Literals. Literals are simply strings, as the title or the time stamps of the program in our example. Nodes can also be blank nodes, which are nodes that neither an URI nor a literal is assigned to. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the nodes in the graph. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations.

[1]The World Wide Web Consortium is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. Led by Web inventor Tim Berners-Lee and CEO Jeffrey Jaffe, W3C's mission is to lead the Web to its full potential. `http://www.w3.org/`

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?book ?title
WHERE {
        ?book abc:hasTitle ?title
}
```

Listing 2.1: Supposing that *exampleOntology* is an ontology which describes a library, then this simple SPARQL query answers the question "What are the titles of all the books in the library?". Variable are indicated by a "?" prefix. $PREFIX$ is a clause which allows to write shorter and clearer queries, using *abc* instead of the complete URI prefix. ?*book* will extract the URI which refers to the concrete object. The result will be all the couples book - title.

Another important aspect to underline with regards to RDF is the easiness of information extraction. SPARQL[2] is the query language developed with this aim. SPARQL stays for SPARQL Protocol And RDF Query Language. It is very similar to the SQL query language, indeed its syntax is very similar: $SELECT$ and $WHERE$ clauses represent in this case the basics too. In the same way SQL was modeled to query databases, SPARQL is developed to query RDF data. For instance, look Listing 2.1 where a very simple query is proposed.

SPARQL allows very specific queries, which will have globally unique results, thanks to the use of the URI's in the RDF which are globally unambiguous. This illustrates the idea of Semantic Web of considering the Web as one big database, the so-called Web of linked data.

SPARQL is only one element of an infrastructure that is necessary to employ in order to accomplish such a task of information retrieval.
First the query needs to be processed by a so-called "SPARQL end-point". This element processes the query inserted by the user and then refers to the data repository in order to satisfy the query. Subsequently, the results are provided to the user, through the end-point.
To accomplish such a task, we employed Cliopatria [16], which is "a SWI-Prolog [34] application that integrates SWI-Prolog's the SWI-Prolog libraries for RDF and HTTP services into a ready to use (semantic) web server". Cliopatria provides the web infrastructure necessary to answer SPARQL queries. It relies on the SWI-Prolog RDF repository, where we stored our network files to be queried.

[2]The right pronunciation is "sparkle".

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE rdf:RDF [
  <!ENTITY rdf  'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY sem 'http://semanticweb.cs.vu.nl/2009/11/sem/'>
  <!ENTITY epg 'http://localhost/EPG_ontology.owl/'>
  <!ENTITY xml 'http://www.w3.org/2001/XMLSchema#'>
  <!ENTITY dbpedia 'http://dbpedia.org/'>
  <!ENTITY dc 'http://purl.org/dc/elements/1.1/'>
  <!ENTITY foaf 'http://xmlns.com/foaf/0.1/'>
]>

<rdf:RDF
  xmlns:rdf="&rdf;"
  xmlns:sem="&sem;"
  xmlns:epg="&epg;"
  xmlns:xml="&xml;"
  xmlns:dbpedia="&dbpedia;"
  xmlns:dc="&dc;"
  xmlns:foaf="&foaf;"
>

<sem:Event rdf:about="http://example.org/program/45465507">
  <sem:hasActor rdf:resource="http://example.org/person/Michael_Kürner"/>
  <sem:hasActor rdf:resource="http://example.org/person/Oliver_Welke"/>
  <sem:hasActor rdf:resource="http://dbpedia.org/resource/Stefan_Raab"/>
  <epg:hasGenre rdf:resource="http://example.org/genre/114.1058"/>
  <epg:hasGenre rdf:resource="http://example.org/genre/122.835"/>
  <epg:hasGenre rdf:resource="http://example.org/genre/88.835"/>
  <epg:hasGenre rdf:resource="http://example.org/genre/98.1033"/>
  <epg:hasGenre rdf:resource="http://example.org/genre/98.324"/>
  <sem:eventType rdf:resource="http://example.org/type/6"/>
  <sem:hasPlace rdf:resource="http://example.org/channel/40"/>
  <epg:hasTitle xml:lang="de">Die TV Total Pokerstars.de-Nacht</epg:hasTitle>
  <sem:hasBeginTimeStamp rdf:resource="2010-04-20T20:10:00.0Z"/>
  <sem:hasEndTimeStamp rdf:resource="2010-04-20T23:10:00.0Z"/>
</sem:Event>

<rdf:Description rdf:about="http://example.org/person/Michael_Kürner"/>
<rdf:Description rdf:about="http://example.org/person/Oliver_Welke"/>
<rdf:Description rdf:about="http://dbpedia.org/resource/Stefan_Raab">
  <dbpedia:birthPlace dbpedia:resource="Köln">
  <dbpedia:birthDate xml:date="1966-10-22">
  <dc:description xml:lang="de">Deutscher Showmaster, Entertainer, Musikproduzent und Musiker</purl:description>
  <rdf:type foaf:Person>
  <foaf:givenName xml:lang="de">Stefan</foaf:givenName>
  <foaf:surname xml:lang="de">Raab</foaf:surname>
  <foaf:name xml:lang="de">Stefan Raab</foaf:name>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/genre/114.1058">
 <rdf:resource xml:lang="de">Poker.Stars<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/genre/122.835">
 <rdf:resource xml:lang="de">Kneipensport.Poker<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/genre/88.835">
 <rdf:resource xml:lang="de">Sport.Poker<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/genre/98.1033">
 <rdf:resource xml:lang="de">Unterhaltung.Spiele<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/genre/98.324">
 <rdf:resource xml:lang="de">Unterhaltung.Event<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/type/6">
  <rdf:resource xml:lang="de">Show<\rdf:resource>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/channel/40">
  <rdf:resource xml:lang="de">ProSieben<\rdf:resource>
</rdf:Description>
</rdf:RDF>
```

Figure 2.5: An example of the EPG network RDF file, encoded in XML format. $< sem : Event >$ is the representation of an episode, while the subsequent triples define the characteristics of the URI's used in the description of the episode.

2.4 Networks Definition

We built two networks, namely:

- an EPG network, where nodes are represented by episodes, genres and guests, and edges link the episodes with their guests and genres.
- a guests network, where nodes are represented by guests, and edges link celebrities that appeared in the same program episode.

For the second network we needed to decide which kind of relation the network will represent. This is a very specific domain point of the definition process. We decided to use the co-participation of the people in the same TV program episode. We could have chosen also other links, such as the participation to the same TV program, without distinguish the different episodes but the relation chosen is, in our opinion, more strong than the latter and fits better our perspective.

Both the networks are build incrementally time wise, *i.e.* for every episode we consider only the network built until that moment (see Figure 2.6). This means that when we take a new episode into consideration, we take the information we need from the network built before the episode start, so we can measure the popularity of guests and genres until that moment in time.

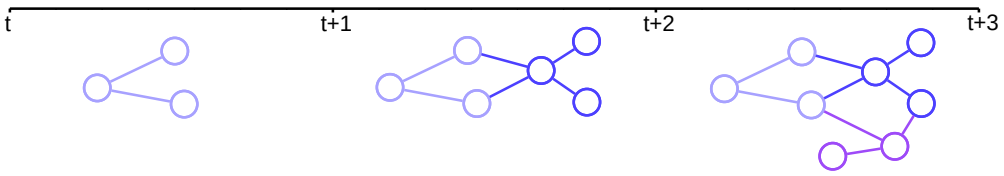

Figure 2.6: The time line of the incremental network

2.4.1 The EPG network

This network is the representation of the EPG as a graph. The aim of this network is to represent the EPG in the past and to link the different TV programs in a new way. Indeed, the TV programs are now connected not only because they have a genre in common, which is a very simple connection, but also through the guests they have in common. And guests themselves are

linked together through the TV programs and through genres too, as Figure 2.7 shows.

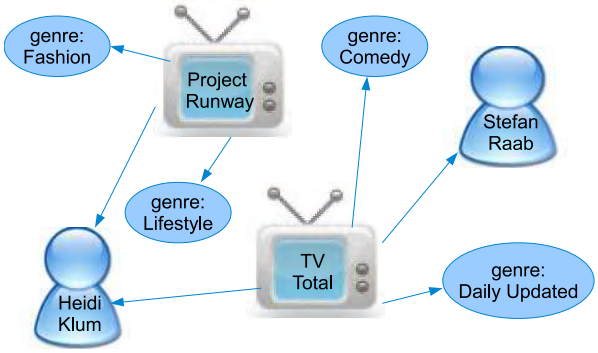

Figure 2.7: A simplified version of the EPG network.

The network was retrieved from the RDF file with the SPARQL in Listing 2.2.

Centrality measure

In the EPG network we use as popularity measure of guests and genres the degree of the vertices with which they are represented.

Definition. *The degree of a vertex into a network is the number of edges it has with the other vertices.*

In terms of our network, this measures how many time guests and genres did occur in all the previous episodes. From our point of view, this measure should help to verify if people that occur many times in the past, *i.e.* with a high degree, are really fundamental in determining the success of a particular episode. The same holds for genres.

```
PREFIX sem: <http://semanticweb.cs.vu.nl/2009/11/sem/>
PREFIX epg: <http://localhost/EPG_ontology.owl>
CONSTRUCT {?x epg:hasTitle ?Title
           ?x sem:hasBegin ?Begin
           ?x epg:hasGenre ?Genre
           ?x sem:hasActor ?Actor}
WHERE {?x epg:hasTitle ?Title
       ?x sem:hasBegin ?Begin
       ?x epg:hasGenre ?Genre
```

```
        ?x sem:hasActor ?Actor}
```

Listing 2.2: Query SPARQL to retrieve the EPG network. *CONSTRUCT* is a clause that gives as result of the query an RDF file. All the words starting with ? are variables name, we could have used any other words. This choice makes the query easily readable.

2.4.2 The social network

Social networks are graph of people which are linked together because they have something in common. For example, the Facebook [14] social network links people because they declared each other to be friends, but also friends of friends are linked through the people they both know. Moreover, people can be linked together through the pages they both liked, and so on. In a similar way, the Twitter [39] social network links people because they "follow" or are "followed" and through the fact that people followed the same person or because they are both followed by the same person. According to Wikipedia:

Definition. *A social network is a social structure made up of individuals (or organizations), which are connected by one or more specific types of interdependency, such as friendship, kinship, common interest, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige.*

In our network, people are linked because they participated to the same TV program episode. Besides, every edge will have a weight, which is the number of times this co-participation occurred (see Figure 2.8).

The social network was retrievied from the RDF file with the SPARQL query in Listing 2.8.

```
PREFIX sem: <http://semanticweb.cs.vu.nl/2009/11/sem/>
SELECT ?actor ?coactor
WHERE { ?id sem:hasActor ?actor
        ?id sem:hasActor ?coactor
        FILTER( ?actor != ?coactor)}
```

Listing 2.3: Query SPARQL to retrieve the social network. The symbol ! = means different. ?*actor* and ?*coactor* are simply variable name, chosen to make the query easier to read.

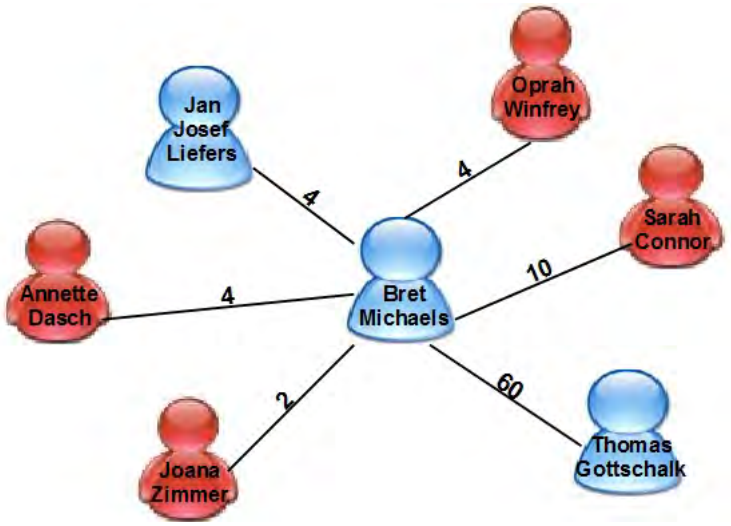

Figure 2.8: A simplified version of the Social network of the TV program *Heute*

Centrality measures

In social network analysis, the centrality measures of a social network are well defined in terms of people popularity. The main contribution is given by [15].

Degree. In social networks, the degree centrality measures the activity level of a person. Generally, it is not always true that "the more connections, the better", usually also where the connections lead to has to be taken into consideration. This is typically true, for example, in the work environment. A person who has many connection with his colleagues, but none with higher spheres, as a "not significant" high degree.

In our situation, "the more connections, the better" holds, since we do not take into consideration the quality of the connections, considering all the guests at the same level.

Betweenness. Betweenness is a centrality measure of a vertex in a graph, defined on the concept of the shortest paths. The shortest path is the minimum number of edges that links two vertices.

Definition. *The betweenness centrality of a vertex is defined as the fraction of all shortest paths that pass through it over all shortest paths in the network.*

That is:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{ST}(v)}{\sigma_{ST}},$$

where σ_{ST} is the number of shortest paths from s to t, and $\sigma_{ST}(v)$ is the number of shortest paths from s to t that pass through a vertex v. In case $\sigma_{ST}(v) = 0$, the betweenness will take 0 as value.

Vertices that occur in many shortest paths between other vertices have higher betweenness. In social networks this centrality measure can be used as an index of the potential of a node (guests) for control of communication. Nodes with high betweenness play a powerful role, but represent necessary ways in the network, which means that without them the network will be divided in two or more parts. So these nodes are crucial for connections. Consider a typical high school situation. Consider the group of sporty and cool guys and the group of nerd boys. Now take into consideration a guy which like sports, but also is good at school. He represents the only connection between the previous two groups. Disregard him for a moment, and the two groups will be not connected at all. That guy has the highest betweenness in the network.

In our social network, guests with high betweenness will not represent a crucial aspect in the prediction.

Pagerank. The PageRank[3] is a variant of the Eigenvector centrality measure. The Eigenvector centrality is a measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.

Definition. *Given four web pages A, B, C, and D, the PageRank, PR of page A can be defined as:*

$$Pr(A) = (1-d) + d\left(\frac{PR(B)}{C(B)} + \frac{(PR(C)}{C(C)} + \frac{PR(D)}{C(D)}\right),$$

[3]The name "PageRank" is a trademark of Google, and the PageRank process has been patented (U.S. Patent 6,285,999[4]). It was developed at Stanford University, to which the patent is assigned.

where d is the damping factor, i.e. the probability that, at any step, a person will continue to surf the Web, which can be set between 0 and 1, and $C(A)$ is the number of outbound links.

In our case we will apply this measure to the social network to determine to more "linked" guests, and to see if the presence of the more linked people to an episode influence the popularity or not.

Chapter 3

Techniques and experiments

In order to make the popularity prediction, we decided to make a binary classification of the dataset, which is divided in two parts, popular and non-popular, based on our popularity measure we define later in this chapter. We used the Bayes algorithm to perform the classification task. In what follows we are going to describe in detail these techniques. Finally, we will describe the different experiments we made in order to improve such predictions.

3.1 Experiment focus

The aim of this thesis is to investigate whether the use of network information is useful for predicting popularity. We performed many experiments, using different points of view, in order to make a complete analysis. In particular we concentrated on the nodes level, since we want to discover possible relations between the guests of a TV program episode and the popularity of the episode and between the characteristics of an episode (genre, type, start time and so on) and its popularity. To grasp the possible interactions we extracted from the networks the centrality measures of nodes and we added them as features for the popularity prediction. In particular, we used well-known interpretation of centrality measures of nodes in social networks (see [15]) as popularity measure of people, in order to show whether popularity of guests influence the popularity prediction.

As features for the prediction, we took into consideration some measure of aggregation of the centrality values, *i.e.* average and variance, but also the maximum, the second highest values and the sum of the most two maximum values. Using these features for the different centrality measures, we aim at

analyzing which measure influence the most the popularity prediction, and at finding a combination of them, which, possibly, improve the prediction.

The average gives a general indication of the centrality values of the guests in an episode. By using the average we want to answer the following questions: is a high (or low) average centrality of the guests of an episode useful to predict the popularity level of the episode?

The variance gives an indication of how uniform the centrality of the different guests in the episode is. Using the variance we want to answer the following question: is the popularity of episodes somehow affected by the homogeneity of the centrality of its guests?

The maximum degree value is used in order to see how the most popular guests influence the popularity of an episode. In particular, we want to answer questions such as: considering only the most popular guests in an episode, can we say something about the popularity of the episode? Is only the most popular guests an important feature for the prediction model?

The second most popular guests is used with a similar aim of the previous feature, only that we consider the second most popular guest instead of the first. This features with also be used together with the previous one.

And, finally, the sum of the two most popular guests is used as a measure of aggregation of the previous two features. This gives a different indication since it does not make any distinctions between the values of the popularity. We can have a very popular guest and an unpopular one, or both very popular, and so on.

3.2 Binary classification

Binary classification is the task of classifying a set of objects into two groups on the basis of whether they have some characteristic or not. More formally, we have a random couple $C(X, Y)$, where $X \in \mathcal{R}^d$ is called the $feature\ vector$ and $Y \in \{0, 1\}$ is called the *label*. The goal is to learn a $classifier$, *i.e.* a map $g : \mathcal{R}^d \to \{0, 1\}$, such that the probability of classification error, $\mathcal{P}(g(X) \neq Y)$, is small.
A standard way of evaluating the quality of a binary classifier is by looking at its probability of classification error. Thus, for a classifier f we define the $classification\ loss$ as:

$$L_p(f) \triangleq \mathcal{P}(f(X) \neq Y) \equiv \int_{X \times \{0,1\}} 1_{\{f(x) \neq y\}} P(dx, dy),$$

where $1_{\{\cdot\}}$ is the *indicator function* taking the value 1 if the statement in the braces is true, and 0 otherwise. We find the best classifier as follows:

Proposition. *Given the joint distribution P on $X \times \{0,1\}$, let $\eta(x) = E[Y|X=x] \equiv \mathcal{P}(Y=1|X=x)$. Then the classifier*

$$f_P^*(x) \triangleq \begin{cases} 1 & \text{if } \eta(x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

minimizes the probability of classification error over all $f : X \to \{0,1\}$, i.e.,

$$L_P(f_P^*) = \min_{f:\to\{0,1\}} L_P(f).$$

Remark. The function η defined above is a *regression function* while the classifier in Equation 3.1 is called *Bayes classifier*.

Proof. Consider an arbitrary classifier $f : X \to \{0,1\}$. Then

$$\begin{aligned} L_P(f) &= \int_{X\times\{0,1\}} 1_{\{f(x)\neq y} P(dx,dy)\} \\ &= \int_X P_X(dx)\{P_{Y|X}(1|x)1_{\{f(x)\neq 1\}} + P_{Y|X}(0|x)1_{\{f(x)\neq 0\}}\} \\ &= \int_X P_X(dx) \underbrace{\{\eta(x)1_{\{f(x)\neq 1\}} + (1-\eta(x))1_{\{f(x)\neq 0\}}\}}_{\triangleq \ell(f,x)} \end{aligned} \tag{3.2}$$

where we have used Equation 3.1, the factorization $P = P_X \times P_{Y|X}$, and the definition of η. From the above, it is easy to see that, in order to minimize $L_P(f)$, it suffices to minimize the term $\ell(f,x)$ in (3.2) separately for each value of $x \in X$. If we let $f(x) = 1$, then $\ell(f,x) = 1 - \eta(x)$, while for $f(x) = 0$ we will have $\ell(f,x) = \eta(x)$. Clearly, we should set $f(x)$ to 1 or 0, depending on whether $1 - \eta(x) \leq \eta(x)$ or not. This yields the rule in (3.1). For further details see [25]. □

3.2.1 Naive Bayes classifier

The Naive Bayes classifier technique is based on the Bayes' theorem. The word "naive" is used to indicate that a strong independence assumption is made to apply such a technique. A Naive Bayes classifier assumes that the attributes are independent (given the class), since the multiplication of probabilities is possible only if the events are independent.

The Naive Bayes probabilistic model

Generally speaking, for a classifier we use a conditional model of the following type:

$$p(C|F_1, \ldots, F_n).$$

This is a conditional model over a dependent class variable C with a small number of outcomes or *classes*, conditional on several feature variables F_1 through F_n. With this kind of probability, we can face two types of problems: (1) the number of features is too big, (2) the possible values the feature can take are too many. Therefore we reformulate the model with the help of the Bayes' theorem:

$$p(C|F_1, \ldots, F_n) = \frac{p(C)p(F_1, \ldots, F_n|C)}{p(F_1, \ldots, F_n)}, \tag{3.3}$$

To be more clear, the above formula can be rewritten as:

$$posterior\ probability = \frac{prior\ probability \times likelihood}{total\ amount\ of\ evidence}.$$

As it can be easily seen, the denominator of equation 3.3 does not depend on the class variable, and the features are already known. Therefore we will concentrate on the numerator, which is equivalent to the joint probability model:

$$p(C, F_1, \ldots, F_n)$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned}
&p(C, F_1, \ldots, F_n) \\
&\quad \propto p(C)p(f_1, \ldots, F_n|C) \\
&\quad \propto p(C)p(F_1|C)p(F_2, \ldots, F_n|C, F_1) \\
&\quad \propto p(C)p(F_1|C)p(F_2|C, F_1)p(F_3, \ldots, F_n|C, F_1, F_2) \\
&\quad \propto p(C)p(F_1|C)p(F_2|C, F_1)p(F_3|C, F_1, F_2) \ldots p(F_n|C, F_1, F_2, F_3, \ldots, F_{n-1}).
\end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every feature F_j for $j \neq i$. This means that

$$p(F_i|C, F_j) = p(F_i|C) \text{ for } i \neq j,$$

and so the joint model can be expressed as

$$p(C, F_1, \ldots, F_n) \propto p(C)p(F_1|C)p(F_2|C)p(F_3|C)\cdots \propto p(C)\prod_{i=1}^{n} p(F_1|C).$$

This means that under the above independence assumptions, the conditional distribution over the class variable C can be expressed like this:

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(F_i|C),$$

where Z (the evidence) is a scaling factor dependent only on $F_1, \ldots, F_n$, *i.e.* a constant if the values of the feature variables are known.
Models of this form are much more manageable, since they factor into a so-called class prior $p(C)$ and independent probability distributions $p(F_i|C)$. If there are k classes and if a model for each $p(F_i|C = c)$ can be expressed in terms of r parameters, then the corresponding Naive Bayes model has $(k-1) + nrk$ parameters [47].

3.2.2 Sampling method

Usually a classification task is applied on the dataset divided into training data and test data. If one is very unlucky, it could happen that the split of the dataset is not fair for both classes. To overcome this possible issue, we applied the 10 Fold Cross-validation method. This sampling method divides the dataset into 10 folds, in which the proportions between the classes of the whole dataset are approximately maintained. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths, then the error rate is calculated on the part held out. Thus, the learning procedure is executed a total of ten times on different training sets. Finally, the 10 error estimates are averaged to yield an overall error estimate.

3.2.3 Evaluation

The evaluation of a binary classification can be done in multiple ways. One of these is to consider the confusion matrix (see Table 3.1). This matrix can be build with the output of the classifier. In particular, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item with the desired one.

	Popular	No popular
Popular	TRUE POSITIVE (tp)	FALSE POSITIVE (fp)
No popular	FALSE NEGATIVE (fn)	TRUE NEGATIVE (tn)

Table 3.1: Confusion matrix: each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

From these numbers we can build two measures, namely precision and recall, which are defined as:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

Recall is the percentage of the true positive classified item, while precision indicate the percentage of positive predicted item correctly classified.

Another indicator that we can extract from the confusion matrix is the *phi coefficient*. This is used in machine learning as a measure of the quality of binary classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The *phi coefficient* is actually a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and $+1$. A coefficient of $+1$ represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. The square of the *phi coefficient* is related to the chi-square statistic for a 2×2 contingency table:

$$\phi^2 = \sqrt{\frac{\chi^2}{n}},$$

where n is the total number of observations. The *phi coefficient* can be calculated directly from the confusion matrix using the formula:

$$\phi = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}.$$

Furthermore, we take into consideration also the ROC area and the overall percentage of correctly classified items.

3.3 Experiment set-up

3.3.1 Popularity measure and definition of the classes

As popularity measure we considered the number of tweets containing the name of the TV program during the episode. The tweets were retrieved using TOPSY API [36,37]. TOPSY is a real-time search engine for the Social Web. This engine indexes and ranks search results based upon the most influential conversations millions of people are having every day about each specific term, topic, page or domain queried. It perfectly fits our requirements, because it allows to retrieve tweets from May 2008 (our older data are of April 2009), while Twitter Search API [1] allows to retrieve, at maximum, one week old tweets.

We use the number of tweets per minute as measure of the popularity. In order to reach this objective, we divided these numbers by the duration of the episodes. To define which programs were popular and which not, we ordered the different episodes by the weighted number of tweets and divided the whole population into two parts: the first part represents non-popular episodes, while the second one the popular episodes.

3.3.2 Feature extraction from the EPG

The data we extracted from the XML files of the EPG are:

- genres: which were converted into a binomial attribute (considering the presence of each single feature value as a binary attribute);
- guests: which were converted into a binomial attribute;
- start time: it was divided in date and time. Additionally, the date was divided in day, month, year and the time was discretized in four periods (Morning (6:01-12:00), Afternoon (12:01-18:00), Evening (18:01-24:00) and Night (00:01-06:00));
- duration of the episode: which was used to weight the number of tweets;
- program type: which was converted into a binomial attribute.

3.3.3 Feature extraction from the networks

As already explained in the previous chapter, we extracted from the networks some interesting centrality measures. Those measures were elaborated in order to being useful features for our prediction model. All the values are normalized applying the logarithm and rounding the value to the nearer integer. From the EPG network we extracted the degree centrality. From the guests network we extracted degree centrality, betweenness centrality and PageRank. Using degree, betweenness and PageRank values we added the following features:

- the mean of the centralities of the guests in the episode: to have an unique measure of the centralities.
- the variance of the centralities of the guests in the episode: to see how much closer (or farther) are the centralities of the guests
- the maximum of the centralities of the guests in the episode: to see if the most central guest is enough to improve the popularity of an episode
- the first two most central guests in the episode: to see if the first two most central guests improve more the popularity of an episode than considering just the most popular
- the sum of the first two most central guests in the episode: to have an unique measure of the first two most central guests in the episode.

3.4 Experiment description

The feature extraction was done with a Java[1] program, while the calculation of the centrality measures was done with an R[2] program. All the experiments were executed with Weka[3], using the *NaiveBayes* algorithm and 10 fold cross validation sampling method.

[1]Java is a programming language and computing platform first released by Sun Microsystems in 1995. It is the underlying technology that powers state-of-the-art programs including utilities, games, and business applications. More information at `http://java.com/en/`

[2]R is a software environment for statistical computing and graphics. Available at `http://www.r-project.org/`

[3]Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka

3.4.1 Experiment on the whole dataset

In order to define a baseline, *i.e.* the starting point from which we will compare the other results, we applied the classification only on the dataset extracted directly from the EPG, namely: guests, genres, start time, and program type. In order to see which were the most important attributes in the classification task, we divided the dataset accordingly, that is, we applied the classification adding a feature at a time. So the classification was performed 5 times, one time for every attribute plus one with all of them. We defined the baseline with the combination that performs better. Starting from this baseline, we added the centralities features one by one. This experiment was performed in order to answer the research hypothesis according to there should be some interaction between the popularity of guests and that of an episode.

3.4.2 Experiment per program type

The same procedure was applied to the dataset divided according to the program type. This additional evaluation was performed in order to underline how different characteristics of different program types influence the behavior of the classifier.

Experiment on the program type *event*

The program type named *event* was selected because it showed the best results among all the program types. We aimed at finding a combination of features that could improve the classification performance. First, we selected the feature for every centrality measures which performed best. Then, also all the other combinations were tested, that is, we performed the classification using every time a different combination of attributes, until we tried all of them out.

3.4.3 Experiment with a different popularity measure

We decided to test whether a popularity measure defined independently from our dataset can perform better in the prediction than the previously defined

contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. More information at `http://www.cs.waikato.ac.nz/ml/weka/`

measures. We used for this purpose the Google counts, *i.e.* how many results Google returns searching for the name of the guest, which can be seen as an example of measure independent from our network, and is particularly significant since it is based on large amounts of observations and it is provided by a reliable service. As feature we used the average of the Google counts of the guests in the episode.

Chapter 4

Results

In this chapter we present the results of our experiments.

4.1 Preliminary analysis

Before starting to apply the Bayes classifier to the data, we made some exploratory analysis in order to identify possible particular patterns that indicate the characteristics of a class. From the graphs it is evident that the two classes, popular and no-popular, have many characteristics in common, which makes difficult to identify clearly the borders of the two classes.

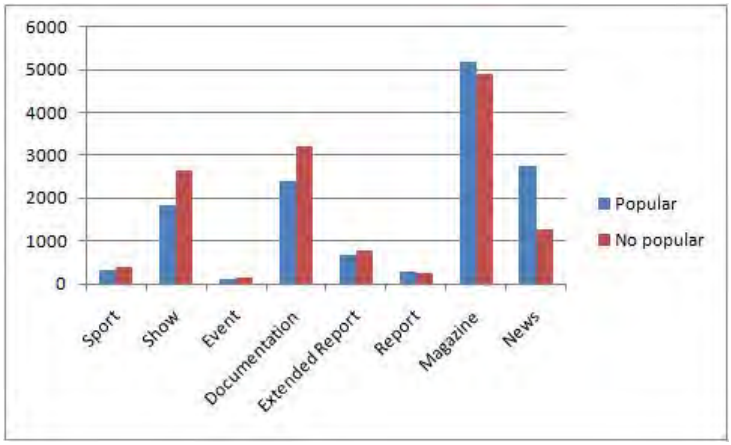

Figure 4.1: The distribution over the dataset of the program types

Additional consideration about the centralities values are:

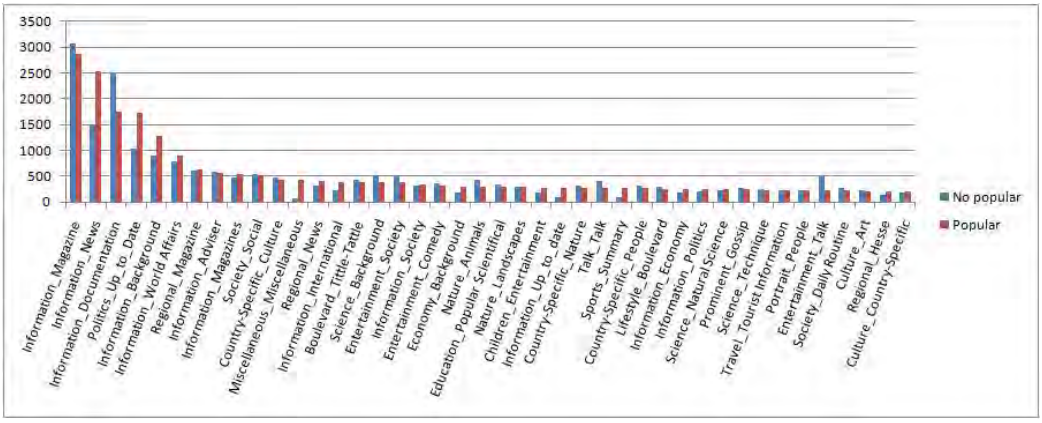

Figure 4.2: The distribution over the dataset of the genres, only genres that appear more then 200 times in the "popular" dataset are considered.

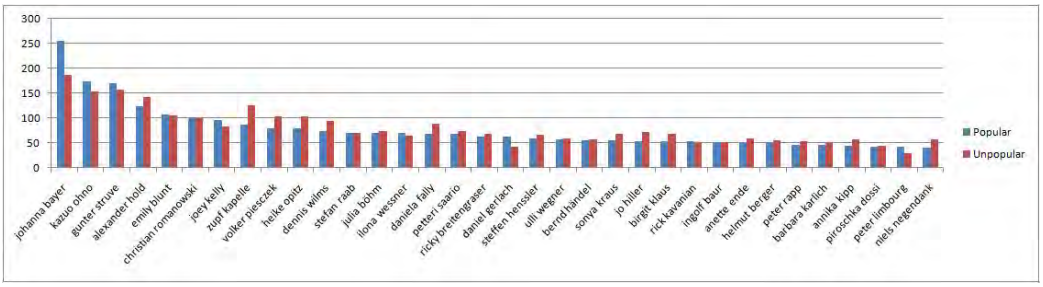

Figure 4.3: The distribution over the dataset of the guests, only guests that appear more then 200 times in the "popular" dataset are considered.

- Popular episodes have lower degree centralities than unpopular ones. This is something that can be explained with the fact that we did not take into consideration the different roles of the people in the episode. This means, for example, that anchormen get the highest degree centralities with respect to guests which appear necessarily less times, although their real-life popularity levels may be inverted.

- Popular episodes have guests with lower betweenness centralities than unpopular ones. This is an interesting aspect and something we actually expected. Since betweenness centrality measures how important a node is for the communication of two sub-graphs, this cannot be really referred to as a popularity measure. Usually nodes with higher betweenness centrality have a low degree.

- Popular episodes have guests with higher PageRank than unpopular ones. With PageRank, also nodes with a low degree centrality can get

a high rank if they are connected to nodes with high degree. Even just one connection with node with an high degree suffices to make it get a high rank. So this means that PageRank could be considered fairer measure, in our particular situation, for the popularity of a guest.

4.2 Baseline Classification

As baseline we used only the data extracted from the simple EPG's, that is genres, guests, start time and program type. Table 4.1 and 4.2 illustrates the results. In Table 4.1 we can observe the results of the classification, in particular we can read precision and recall. As reminder, recall is the percentage of the true positive classified episodes, while precision indicate the percentage of the positive predicted episodes correctly classified. In Table 4.2 we can see other results of the same classification. In particular there are reported the ROC area, the Phi coefficient and the overall percentage of correctly classified episodes. The ROC area is the area below the ROC curve, which is obtained drawing the true positive against the false positive rate. The Phi coefficient is a correlation coefficient between the observed and predicted binary classifications: it returns a value between -1 and 1, where -1 indicates an inverse prediction, while 1 indicates a perfect prediction.

The best results are obtained when making an attribute selection for guests and genres plus the program type and the start time. The attribute selection was made on the basis of the frequency of the attribute: only genres and guests which appear at least 200 times in the dataset are taken into consideration.

As a general remark, looking at both tables we can say that the most powerful features in the prediction are the genres, while the least powerful are the guests. This is not exactly what we expected, since our idea was that the more popular is a guest, the more popular an episode that hosts him will be. In other words what we thought was that a very popular person is followed by his or her fans everywhere, no matter the TV program they appear in. However, these results seem not to confirm our idea.

4.3 Classification with networks property

In order to analyze the influence of each centrality measure extracted from the network on the predictor behavior, we added separately every new fea-

Dataset	Class	Precision	Recall
Popular genres and guests, start time, program type	Popular	0.639	0.393
	No Popular	0.562	0.778
Only popular genres	Popular	0.634	0.369
	No Popular	0.555	0.787
Only popular guests	Popular	0.52	0.268
	No Popular	0.507	0.753
Only programs types	Popular	0.562	0.608
	No Popular	0.574	0.527
Only start time	Popular	0.536	0.323
	No Popular	0.516	0.721

Table 4.1: Precision and recall of the baseline classification.

Dataset	ROC Area	Phi coefficient	% Correctly classified
Popular genres and guests, start time, program type	0.623	0.135	58.58%
Only popular genres	0.614	0.122	57.81%
Only popular guests	0.515	0.017	51.07%
Only programs types	0.585	0.132	56.75%
Only start time	0.531	0.036	52.19%

Table 4.2: Area below the ROC curve, Phi coefficient and percentage of correctly classified instances of the baseline classification.

ture, as defined in the previous chapter. Namely we tried out the average, the maximum, the variance, the sum of the first most central value, the first two most central value for degree, betweenness and PageRank. In Tables 4.3 and 4.4 we reported only the best results. In this case, looking at the different evaluation measures, we can see no accordance between them. In particular, looking at precision (Table 4.3), the best results are reached using the average of the betweenness centrality. These results are a little bit surprising, especially considering that betweenness can be seen as an index of the potential of a guest for control of communication, this does not seem

an important feature for the popularity prediction. However, looking at Table 4.4, if we take into consideration the ROC area and the Phi coefficient, the best results are reached using the variance of PageRank, which seems a more reasonable results, since the PageRank gives an indication of the quality of the connections a guest has. While looking at the overall percentage of correctly classified episodes, again the best results are reached with the betweenness.

Dataset	Class	Precision	Recall
Baseline + Degree Centrality (average)	Popular	0.640	0.395
	No Popular	0.563	0.779
Baseline + Betweenness Centrality (average)	Popular	0.643	0.396
	No Popular	0.564	0.780
Baseline + PageRank (variance)	Popular	0.632	0.419
	No Popular	0.566	0.757

Table 4.3: Precision and recall, best results on the full dataset.

Dataset	ROC Area	Phi coefficient	% Correctly classified
Baseline + Degree Centrality (average)	0.624	0.136	58.67%
Baseline + Betweenness Centrality (average)	0.624	0.138	58.78%
Baseline + PageRank (variance)	0.625	0.140	58.77%

Table 4.4: Area below the ROC curve, Phi coefficient and percentage of correctly classified instances, best results on the full dataset.

As an overall remark, we can say that these results show a very poor influence of the centrality measures on the predictor behavior.

4.3.1 Results per program type

We applied our analysis on a dataset composed by TV programs of different types, which have, of course, different characteristics. Until now we showed the results on the overall dataset. Now we will show the results of the analysis

applied to the dataset divided on the basis of the programs type. The aim of this division is to underline that, given the right prerequisites, our approach could give interesting results.

Tables 4.5, 4.6 and 4.7 report the results divided by different centrality measures. In particular, in the second column it is always reported the overall percentage of correctly classified episodes with the classification performed with the baseline. In the remaining 5 columns, the same measure is reported adding one by one to the baseline the different features we built with the centrality values of the episodes.

Figures 4.4, 4.5 and 4.6 represent the graphs where a comparison between the classification performance is made. In particular such graphs report the difference between the classification results performed with the baseline, and the classification results performed with the baseline plus the new feature. A bar on the left indicates a better performance of the baseline, while a bar on the right indicates a better performance of the added new feature.

Program type	Baseline	Degree Centralities				
		Average	Maximum	Variance	Sum first 2	First 2
Sport	63.25%	61.60%	62.05%	62.05%	62.05%	61.75%
Show	59.06%	59.24%	59.35%	58.85%	59.35%	59.31%
Event	61.00%	61.47%	61.47%	63.30%	61.00%	63.76%
Documentation	58.00%	58.00%	58.10%	57.62%	58.06%	57.86%
Extended report	57.44%	57.22%	57.30%	57.08%	57.22%	57.08%
Report	65.90%	66.47%	66.47%	66.09%	66.28%	66.67%
Magazine	57.45%	57.50%	57.42%	57.47%	57.44%	57.39%
News	65.98%	66.17%	66.10%	66.10%	66.15%	65.77%

Table 4.5: Degree centralities performances per program type.

Looking especially to Figures 4.4, 4.5 and 4.6, we can see that the degree centrality and the PageRank centrality do classify better than the baseline the most of the times and with a significant difference, while the betweenness centrality classifies better than baseline few times and with a very low difference. This confirms our hypothesis according to the betweenness centrality cannot be a good feature for the popularity prediction.

Furthermore, these results show that for a specific program type, namely the program type *"event"*, our approach gives very good results. This probably due to the fact that this program type fits very well our prerequisites:

- guests are really important for this type of programs;

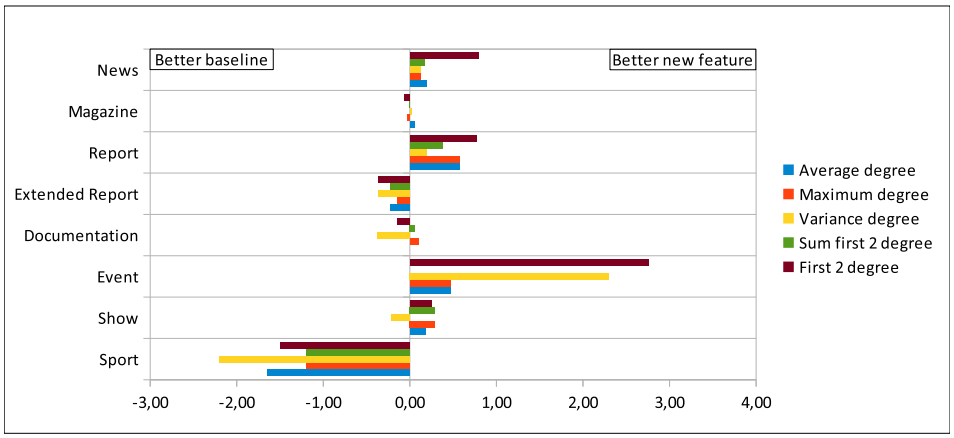

Figure 4.4: Differences between the classification performed with the baseline and with the degree centralities.

Program type	Baseline	PageRank				
		Average	Maximum	Variance	Sum first 2	First 2
Sport	63.25%	63.70%	61.75%	64.16%	62.80%	62.35%
Show	59.06%	58.76%	59.01%	59.03%	58.76%	59.00%
Event	61.00%	63.00%	60.10%	60.55%	61.93%	61.47%
Documentation	58.00%	57.97%	58.00%	58.12%	58.00%	58.06%
Extended report	57.44%	57.67%	58.00%	58.12%	58.06%	58.00%
Report	65.90%	66.09%	67.44%	66.09%	66.09%	66.09%
Magazine	57.45%	57.47%	57.62%	57.62%	57.49%	57.56%
News	65.98%	65.85%	66.10%	66.10%	65.82%	65.95%

Table 4.6: PageRank performances per program type.

Program type	Baseline	Betweenness Centrality				
		Average	Maximum	Variance	Sum first 2	First 2
Sport	63.25%	62.20%	63.25%	63.25%	63.40%	62.65%
Show	59.06%	58.54%	58.31%	58.35%	58.38%	56.68%
Event	61.00%	61.47%	59.17%	60.09%	60.10%	52.29%
Documentation	58.00%	58.06%	58.17%	58.25%	58.17%	57.43%
Extended report	57.44%	57.15%	57.22%	57.45%	57.22%	55.67%
Report	65.90%	65.90%	62.28%	66.09%	66.47%	65.32%
Magazine	57.45%	57.58%	57.37%	57.42%	57.31%	57.73%
News	65.98%	66.10%	65.85%	65.72%	66.05%	66.65%

Table 4.7: Betweenness centralities performances per program type.

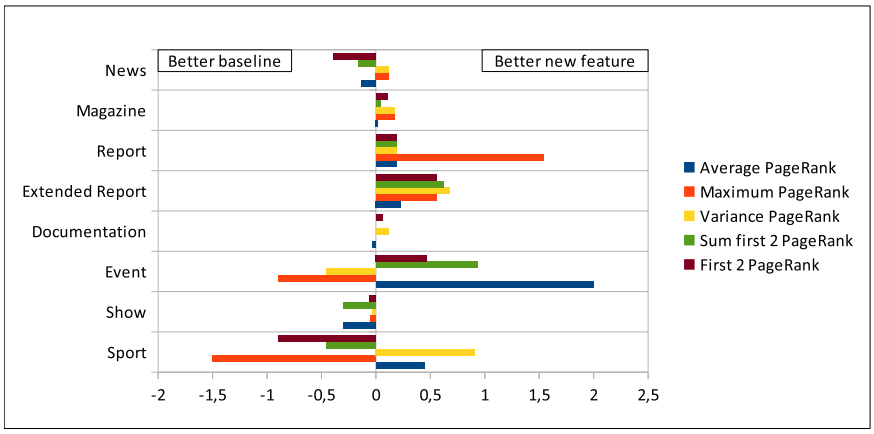

Figure 4.5: Differences between the classification performed with the baseline and with the Pagerank values.

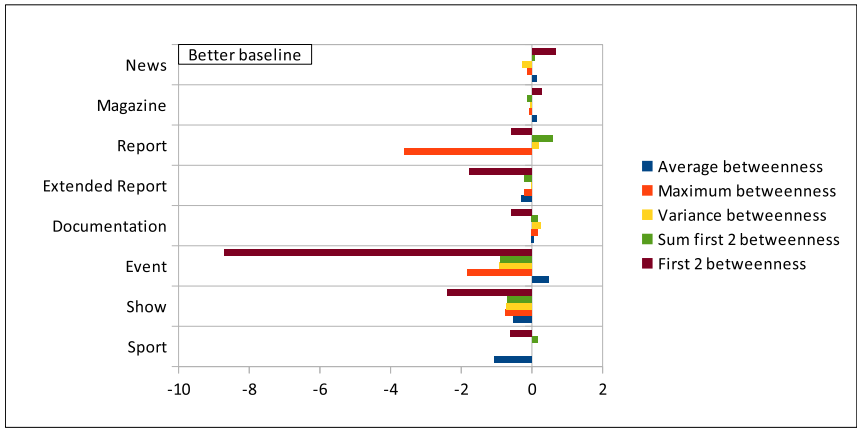

Figure 4.6: Differences between the classification performed with the baseline and with the betweenness centralities.

- expectation for this type of programs stimulates the buzz;

- title of such programs are more recognizable in the tweets.

For the other program types, the results are quite the same as before. In the next section we will focus our attention on the program type *"event"*, to which we applied more detail analyses.

4.3.2 Program type *"event"*

Now we concentrate the analysis on the program type *"event"*. In particular we concentrate our attention on the possibility to improve the results combining different features. As first trial, we add all the features which give the best results, namely the first two most central values of the degree centrality, the average of PageRank and the average of the betweenness centrality. The percentage of correctly classified instances increases to 64.22%. However the highest result is reached when adding the average, the variance and the second most high value of the degree centrality. The percentage of correctly classified instances increases to 66.97%. Adding other features does not improve the classification any more. Figure 4.7 shows the differences between the results of the classification performed with the baseline and the classification results performed with the baseline plus the new feature.

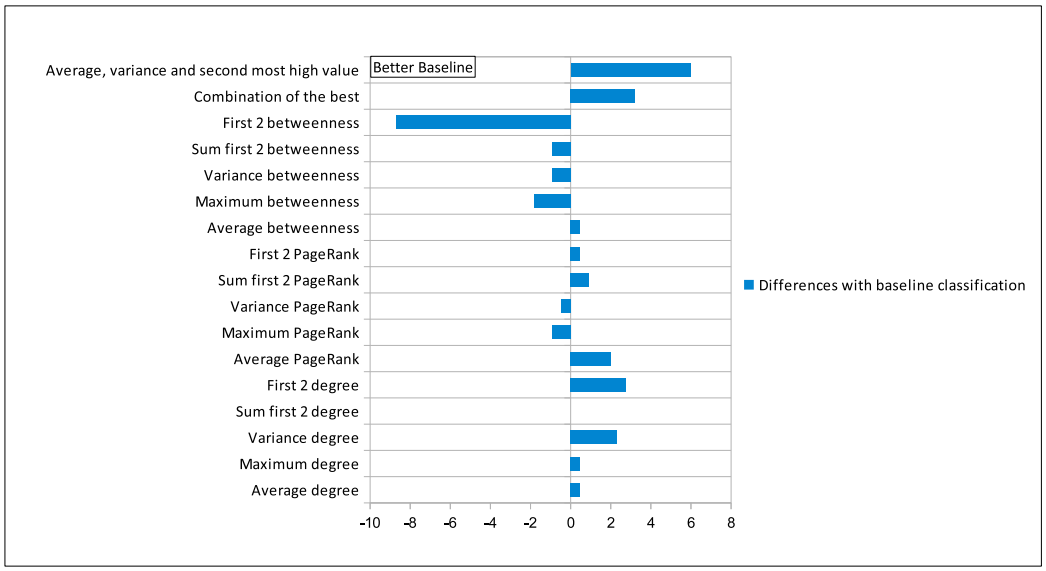

Figure 4.7: Differences between the classification performed with the baseline and with the event program type centralities.

The classification with the new features performs a better classification than the baseline the most of the time. In particular, degree centrality and PageRank correctly classify more often. Betweenness centrality performs a correct classification less often. So, also these results confirm our hypothesis according to the betweenness centrality cannot be a good feature for the popularity prediction.

4.4 A different popularity measure for guests

We take into consideration Google counts as a different popularity approximation for the guests. This option is explored in order to see if an approximation of the popularity measure defined from an external source can improve the results. In particular we want to define a popularity measure which can better reflect the real life popularity, independently from our network.

Program type	Baseline	Google counts
Sport	63.25%	65.81%
Show	59.06%	58.76%
Event	61.00%	61.00%
Documentation	58.00%	57.55%
Extended report	57.44%	57.72%
Report	65.90%	66.67%
Magazine	57.45%	57.63%
News	65.98%	65.85%

Table 4.8: Classification with Google counts.

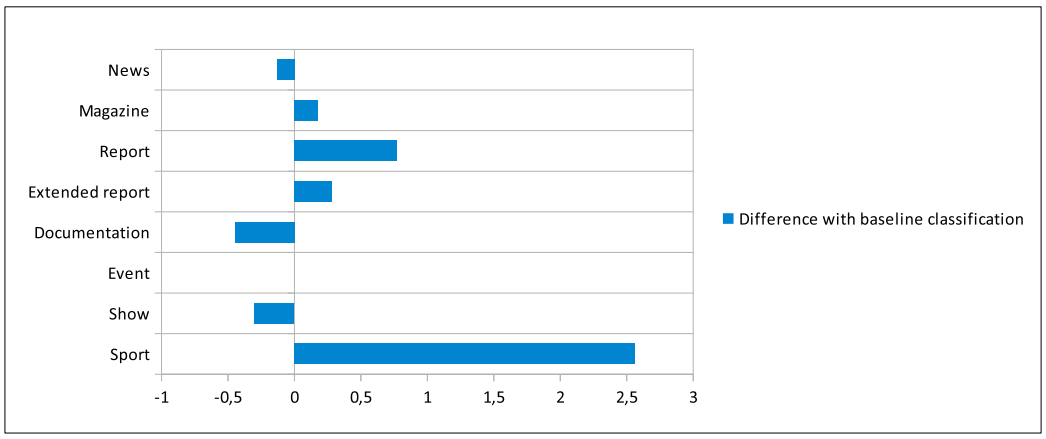

Figure 4.8: Differences between the classification performed with the baseline and with the Google counts values.

Table 4.8 reports the overall percentage of correctly classified episodes with the classification performed with the baseline and with the added new feature. Figure 4.8 is the graph of the diferences between the two classification results.

As Table 4.8 and Figure 4.8 show, Google counts give a significant result with the program type *"sport"*, for which the classification is improved by

2.56%. First we have to say that these are programs which have only the anchorman in the episode, so using the mean is the same as using the value of the Google counts. Also, there is a very strict distinction between anchormen of popular episodes and anchormen of unpopular ones. So the improvement in the classification is probably due to the fact that Google counts make this distinction even more marked.

Chapter 5

Conclusion and future work

In this Chapter, we draw the conclusions from what we have seen from our analyses, and we give an indication for further investigations.

5.1 Conclusions

We performed our analyses on two levels. First we applied the analyses on the whole dataset, and then we compare the results considering the different program types.

The results using the entire dataset seem to indicate that no correlation is present between the popularity of an episode, as we defined it, and the centralities we extracted from the networks, *i.e.* our approximation for the popularity of guests. It is likely that multiple reasons are the causes of these poor results. First, we must recall that, although it is correct to assume that people tweet about TV programs, not every program is uniquely identified by a corresponding *hashtag*. In particular, the TV programs that we analyzed do not really promote the use of Twitter, as, for instance, BBC does, putting the hashtag of the program on the screen. Therefore, having decided to take an uniform approach on our measurements, we chose to recognize the title of the TV program within the tweets. Hence, our results will carry some unavoidable imprecisions, due to the difficulty of this task. For instance, we find a lot of tweets containing the word "wetter" which is a title of a program and means "weather". However, it is unlikely that all those tweets were actually referring to that TV program. Furthermore, some programs have a title composed by many words. Usually, the hashtag of a specific TV program would be the result of a combination of the words in the title,

linked, for instance, by hyphens or underscores, plus other words, such as *live*. So, even guessing the real hashtag of a TV program would be very difficult, while identifying the name of the program in the tweet without ambiguities, is even harder. Another reason why we have poor results is that, because of the non-uniformity of TV program descriptions, we could not recognize the roles of the people participating in a TV program from all the descriptions. Therefore, we could not incorporate that kind of information in the networks we built and thus, the centrality values we obtained do not always approximate the real life popularity of the guests. We could have made a better approximation by taking into consideration the different roles of the people. By having that kind of information at disposal, we would have build a directed network, and used in-degree as popularity measure of guests. For instance, anchormen have a very high degree centrality in our network. However they should have an high out-degree and a low in-degree, since they always appear in the program but they are the inviting people, not the invited.

Analyzing the data considering the different program type, we find that actually there is one program type in the dataset that fits well our prerequisites, namely the program type *event*. In particular, this program type enhances the "buzz" around it. This kind of programs goes on air only few times during the year, even just once. So they create around them a kind of expectation and people are really waiting for them. These are kind of programs that call especially for gossip around them, as, for instance, the Oscar Award Ceremony mentioned in the first chapter does. Furthermore, these programs usually have titles which are more recognizable in the Twitter cloud, because they are composed by unusual noun in the every-day speaking. This allows to overcome the issue about the search of the *hashtag* of the TV programs, making the count of the tweets a more reliable popularity measure.

Finally, these are kind of programs for which the guests really make the difference. Consider again the Oscar Ceremony Awards. On Twitter, people was not really speaking about the prizes, but only about how did a person behave or how that actress was dressed and so on. So people tweet mainly about the people, and not about the program in itself. Guests are fundamental to create the buzz.

The analyses on this type of programs underlines the presence of a correlation between the degree of a guest and the popularity of an episode. In particular, the most influent features on the predictor are the average of the degree of all the guests, the variance of the degree of all the guests and the guest with the second degree value. We analyze these features one by one.

Average is a fair measure of the degree centrality with respect to the total number of guests in the episode. On average, popular episodes have lower average degree of guests than unpopular ones. This can be explained as follows: high average can indicate the presence of an anchorman, which has an high degree, and few or no guests, which have low degree, while a low average can indicate the presence of many guests, with a low degree, and a presence of an anchorman with a low degree. This indicates that a person popular in our network, *i.e.* with an high degree, is not a so popular person in real life. So, it is likely that guests with low degree in our network are more popular in real life than guests with high degree, and so the degree centrality should be read as the inverse of the popularity of a person in real life.

Variance is a measure of the inverse of the uniformity of the degrees of the guests in an episode, *i.e.* low variance indicates high uniformity. Popular episodes have on average lower variance than unpopular ones. For instance, when the only person participating in a TV program is the anchorman, the variance will be low, because the single degree is clearly "uniform" with itself. The same holds when the episode hosts many people, all having high (or low centrality). On the other hand, when the anchorman (supposing, as before, that he holds an high degree) participates to an episode with guests having low degree, then the variance of the degrees of the programs rises. From our analyses it seems that the uniformity of the popularity of guests in an episode determines its popularity.

Finally, consider the guest with the second degree value. This is a more powerful feature than the highest degree value, since the differences between the two classes, popular and unpopular, are more marked. This leads again to the consideration about the anchormen. In particular this underlines the fact that the most central guest is not fundamental for the popularity of an episode. Since it is likely that the most central guest is the anchorman, only looking at the second most central guest, we can have a better indication about the degree of the "real" guests. Furthermore, taking into consideration the fact that, on average, popular episodes have guests with lower degree than unpopular ones, it is possible to conclude that this is a crucial value to determine the popularity of an episode.

We think that the contribution of this work is twofold:

- First, a novel representation for EPG is proposed: using SEM we propose a very simple way of represent TV programs information keeping the point of view of the watcher. This can be useful to integrate in-

formation from other sources in the guide, but also to enhance other analyses of this kind.

- Second, this work underlines the presence of a correlation between Social Networks and TV programs popularity. Hence, this work represents the first step towards a more detailed analysis of these interactions.

5.2 Future work

Possible extensions of this work can follow different directions. First, the role of the people in the show should be taken into consideration. This will allow to build a directed graph and so, to measure in-degree and out-degree and be more precise about the popularity of guests. However, as we already underlined, this will be a difficult task, since not always the roles of the guests are clearly stated. Second, an extensive research for the real hashtags of the TV programs can help in defining a more reliable popularity measure. Furthermore, fusing information from different Social Networks, we can define even a more reliable measure. Finally, to exploit all the advantages given by the use of RDF, we can integrate information from different sources in our network. For instance, integrating Wikipedia pages of the guests, it is possible to define more relations between them, and extract a different guest network.

Bibliography

[1] Twitter API. [Online] https://dev.twitter.com/, August 2011.

[2] A. Scherp, T. Franz, C. Saathoff and S. Staab. F-a model of events based on the foundational ontology dolce+DnS ultralight. In *Proceedings of the fifth international conference on Knowledge capture (K-CAP '09)*, pages 137–144, 2009.

[3] Alexia Tsotsis. The Oscars, On Twitter: Over 1.2 Million Tweets, 388K Users Tweeting. [Online] http://techcrunch.com/2011/02/28/the-oscars-twitter, August 2011.

[4] alias-i. Lingpipe. [Online] http://alias-i.com/lingpipe/, August 2011.

[5] alias-i. Named Entity Tutorial. [Online] http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html, August 2011.

[6] Appozite,LLC. Tweetreach. [Online] http://www.tweetreach.com, August 2011.

[7] S. Azur and B. A. Huberman. Predicting the Future with Social Media. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 1:492–499, 2010.

[8] B. Lui, D. K. W. Chiu, Haiyang Hu and Hua Hu and Yi Zhuang. Ontology Based Content Management for Digital Television Services. In *e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on*, pages 565–570, October 2009.

[9] BBC. Programmes Ontology. [Online] http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml, August 2011.

[10] W. M. Bulkeley. TR10: Social TV. [Online] http://www.technologyreview.com/communications/25084/, 2010.

[11] Digg Inc. Digg. [Online] http://www.digg.com, August 2011.

[12] E. K. Wells. 3,600 Tweets for EastEnders : Social TV Stats in the UK. [Online] http://www.tvgenius.net/blog/2011/03/21/3600-tweets-eastenders-social-tv-uk, March 2011.

[13] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. [Online] http://www.w3.org/TR/rdf-sparql-query/, August 2011.

[14] Facebook. [Online] www.facebook.com, August 2011.

[15] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[16] G. Schreiber, A. Amin, M. van Assem, V. de Boer, L. Hardman, M. Hildebrand, L. Hollink, Z. Huang, J. van Kersen, M. de Niet, B. Omelayenko, J. van Ossenbruggen, R. Siebes, J. Taekema, J. Wielemaker, and B. Wielinga. Multimedian e-culture demonstrator. pages 951–958, 2006.

[17] G. Szabo and B.A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, August 2010.

[18] J. G. Lee, S. Moon, and K. Salamatian. An Approach to Model and Predict the Popularity of Online Contents with Explanatory Factors. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1 of *WI10*, pages 623–630, Toronto, Canada, 2010.

[19] J. Kim and S. Kang. An ontology-based personalized target advertisement system on interactive TV. In *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pages 895–896, January 2011.

[20] K. Kaneiwa, M. Iwazu and K. Fukuda. An upper ontology for event classifications and relations. In *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence (AI'07)*, pages 394–403, Berlin, Heidelberg, 2007.

[21] K. O'Connor. How Social Media Is Changing Television As We Know It. [Online] http://www.uxbooth.com/blog/how-social-media-is-changing-television-as-we-know-it/, May 2011.

[22] Richard Kastelein. Endemol CEO - "Social TV is Going to Be Huge". [Online] http://www.appmarket.tv/opinion/1003-endemol-ceo-qsocial-tv-is-going-to-be-hugeq.html, August 2011.

[23] V. Krebs. The Social Life of Routers. *Internet Protocol Journal*, 3:14–25, December 2000.

[24] C. Lawton. TV + Social Network = ? [Online] http://online.wsj.com/article/SB122461909287855339.htm, October 2008.

[25] M. Raginsky. Introduction: What is Statistical Learning Theory. [ONLINE] http://people.ee.duke.edu/~maxim/teaching/spring11/notes/intro.pdf, August 2011.

[26] Machine Learning Group at University of Waikato. Weka. [Online] http://www.cs.waikato.ac.nz/ml/weka/, August 2011.

[27] N. Satry, E. Yoneki and J. Crowcroft. Buzztraq: predicting geographical access patterns of social cascades using social networks. In *In ACM EuroSys SNS Workshop (SNS09)*, Nuremberg, Germany, 2009.

[28] N. Willems, W. R. van Hage and J. Janssens. An Integrated Approach for Visual Analysis of a Multi-Source Moving Objects Knowledge Base. *IJGIS10*, 24(10), October 2010.

[29] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of LREC 2010 Map of Language Resources, Technologies and Evaluation*, 2010.

[30] R. Shaw, R. Troncy and L. Hardman. LODE: Linking Open Descriptions of Events. In Gómez-Pérez, Asunción and Yu, Yong and Ding, Ying, editor, *The Semantic Web*, volume 5926 of *Lecture Notes in Computer Science*, pages 153–167. 2009.

[31] S. Bechhofer and A. Miles. SKOS Simple Knowledge Organization System Reference. [Online] http://www.w3.org/TR/skos-reference/, August 2011. W3C, W3C Recommendation.

[32] S. Jamali and H. Rangwala. Digging Digg: Comment Mining, Popularity Prediction, and Social Network Analysis. In *Proceedings of the 2009 International Conference on Web Information Systems and Mining (SNS09)*, WISM09, pages 32–38, Washington, DC, USA, 2009.

[33] S. Wang and P. Groth. Measuring the dynamic bi-directional influence between content and social networks. In *9th International Semantic Web Conference*, ISWC2010, Shanghai, China, November 2010.

[34] SWI-Prolog. [Online] `http://www.swi-prolog.org`, August 2011.

[35] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[36] Topsy Labs, Inc. Topsy. [Online] `http://topsy.com/`, August 2011.

[37] Topsy Labs, Inc. Topsy API. [Online] `http://corp.topsy.com/developers/api/`, August 2011.

[38] TV Genius Ltd. TV Genius. [Online] `http://www.tvgenius.net`, August 2011.

[39] Inc Twitter. Twitter. [Online] `http://www.twitter.com`, August 2011.

[40] W. Kibbe and L. Schriml. Disease Ontology. [Online] `http://do-wiki.nubic.northwestern.edu/index.php/Main_Page`, August 2011.

[41] W. R. van Hage, V. Malaisé, G. de Vries, G. Schreiber and M. van Someren. Combining ship trajectories and semantics with the simple event model (sem). In *Proceedings of the 1st ACM international workshop on Events in multimedia (ACM EiMM09)*, pages 73–80, New York, NY, USA, 2009.

[42] W. R. van Hage, V. Malaisé, G. De Vries, M. Van Someren and G. Schreiber. Abstracting and Reasoning over Ship Trajectories and Web Data with the Simple Event Model (SEM). *MTAP11*, January 2011.

[43] W. R. van Hage, V. Malaisé, R. Segers, L. Hollink and G. Schreiber. Design and use of the Simple Event Model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web*, April 2011.

[44] W3C OWL Working Group. OWL 2 Web Ontology Language. [Online] http://www.w3.org/TR/owl2-overview/, August 2011. W3C, W3C Recommendation.

[45] W3C RDF Working Group. Resource Description Framework (RDF). [Online] `http://www.w3.org/RDF`, August 2011.

[46] W3C XML Working Group. Extensible Markup Language (XML). [Online] `http://www.w3.org/XML/`, August 2011.

[47] Wikipedia. Naïve Bayes classifier. [Online] http://en.wikipedia.org/wiki/Naive_Bayes_classifier, August 2011.

[48] Wikipedia. Graph (mathematics). [Online] http://en.wikipedia.org/wiki/Graph_(mathematics), August 2011.

[49] Y. Mizoguchi-Shimogori, T. Nakamoto, K. Asakawa, S. Nagano, M. Inaba and T. Kawamura. TV navigation agent for measuring semantic similarity between programs. In *Proceedings of the 2007 OTM confederated international conference on On the move to meaningful internet systems - Volume Part I*, pages 75–84, 2007.

[50] Y. Raimond and S. A. Abdallah. The event ontology. [Online] http://purl.org/NET/c4dm/event.owl, 2006.

[51] YouTube, LLC. YouTube. [Online] http://www.youtube.com, August 2011.