FIRST SOCIAL NETWORK ANALYSIS TRUST & PEOPLE SEARCH TECHNIQUES

SmartH2O

Project FP7-ICT-619172

Deliverable D4.2 WP4

Deliverable
Version 1.1 – 26 June 2015
Document. ref.:
D42.POLIMI.WP4.V1.1

Programme Name:	ICT
Project Number:	619172
Project Title:	SmartH2O
Partners:	Coordinator: SUPSI Contractors: POLIMI, UoM, SETMOB, EIPCM, TWUL, SES, MOONSUB
Document Number:	smarth2o. D4.2.POLIMI.WP4.V0.1
Work-Package:	WP4
Deliverable Type:	Document
Contractual Date of Delivery:	30 June 2015
Actual Date of Delivery:	30 June 2015
Title of Document:	First Social Network Analysis trust & people search techniques
Author(s):	Piero Fraternali, Luca Galli, Carlo Bernaschina, Eleonora Ciceri, Matteo Giuliani, Andrea Cominola, Simona Denaro, Ahmad Alsahaf, Bojana Bislimovska, Alessandro Facchini, Evi Lazaridou, Jasminko Novak
Approval of this report	June 29 2015
Summary of this report:	This document contains a review of existing social network analysis, online game player behavioural analysis, and trust and people search techniques; it also surveys the adversarial user's behaviour detection methods, employed in social games to detect malicious behaviours; it proposes an evaluation of the techniques in the abovementioned categories applicable for SmartH2O, with focus on the specificities of both large and uncontrolled deployment scenario and smaller scale and more controlled user bases.
History:	see version history table
Keyword List:	social network analysis, influence detection, adversarial behaviour detection, online player behavioural analysis

Availability This report is public

This work is partially funded by the EU under grant ICT-FP7-619172

Disclaimer

This document contains confidential information in the form of the SmartH2O project findings, work and products and its use is strictly regulated by the SmartH2O Consortium Agreement and by Contract no. FP7- ICT-619172.

Neither the SmartH2O Consortium nor any of its officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2013-11) under grant agreement n° 619172.

The contents of this document are the sole responsibility of the SmartH2O consortium and can in no way be taken to reflect the views of the European Union.

Document History

Version	Date	Reason	Revised by
0.1	04/05/2015	DDP ready	P. Fraternali
0.2	08/05/2015	Section 2 (PMI, EIPCM)	P. Fraternali
0.3	14/05/2015	Added Section 3 (MSM)	P. Fraternali
0.4	19/05/2015	Added Section 4 (PMI)	P. Fraternali
0.5	21/05/2015	Intro and exec summary (PMI)	P. Fraternali
0.6	03/06/2015	Extended bibliography (all)	P. Fraternali
0.7	10/06/2015	Revised Sec 2 (EIPCM, PMI)	P. Fraternali
0.8	23/06/2015	Added Section 5 (PMI, SUPSI)	P. Fraternali
0.9	24/06/2015	Added Section 6 and conclusions	P. Fraternali
1.0	25/06/2015	Revised evaluation in sections 2, 3, 4, revised references and formatting	P. Fraternali
1.1	26/06/2015	Final version for QA	A. Castelletti, A. E. Rizzoli

Table of Contents

Executive Summary

This document is the Deliverable **D4.2, FIRST SOCIAL NETWORK ANALYSIS TRUST & PEOPLE SEARCH TECHNIQUES**, which, according to the DoW has the following goals:

This deliverable contains a review of existing social network analysis, online game player behavioural analysis, and trust and people search techniques; it also surveys the adversarial user's behaviour detection methods, employed in social games to detect malicious behaviours, such as cheating and spamming; it proposes an evaluation of the techniques in the abovementioned categories applicable for the SmartH2O water consumers' networks, with focus on the specificities of both large and uncontrolled deployment scenarios and of the smaller scale and more controlled user base, typical of the variety of the potential application scenarios to be encountered in the real world.

The deliverable is organised as follows:

- **Section 1** provides an introduction, recalling why social network and behavioural analysis are relevant to the SmartH2O project.
- **Section 2** addresses the very broad topic of social network analysis (SNA), which has a long tradition and a vast mass of scientific literature. It starts with the basic tools of graph modelling, management and mining (Section 2.1), which are employed in most SNA works. Next, it concentrates on the specific problem of community and role detection (Section 2.2), which aims at investigating the internal structure of an online community and the various roles that users play within such structure. Finally, it surveys the field of influence and trust computation, which targets the detection of users with special role in the community and is thus instrumental to the optimization of several community interactions, e.g., the spread of information or the boosting of engagement in the social network.
- **Section 3** surveys the field of online game behavioural analysis, which is concerned with the collection and analysis of activity traces by game players, in order to detect as early as possible those users that evade the game rules in order to acquire an undue advantage. The section provides a set of six requirements for the applicability of the online game behavioural analysis and contrasts the main approaches based on their adequacy with respect to the stated requirements.
- **Section 4** zooms into the specific problem of adversarial behaviour detection; here the online game behavioural analysis becomes more specific: users not only “play”, but in doing so also contribute some input that the system must exploit. The section thus reviews and evaluates a sample of the most relevant techniques that can be applied in order to maximize the confidence in the user’s input.
- **Section 5** discusses how the social network analysis and user behaviour techniques illustrated in Section 2, 3 and 4 integrate with the data mining and user modelling approach of SmartH2O. The rational is the exploitation of an original data fusion method to user modelling, which merges two independently collected data streams (social awareness data and water consumption data) in order to build a more accurate and hopefully representative model of the user’s behaviour.
- **Section 6** provides a conclusive summary of all the pertinent social awareness techniques, which have been previously illustrated and evaluated independently of the requirements of the SmartH2O project; it provides their positioning within the SmartH2O requirements and architecture and, in doing so, evaluates the more relevant techniques for SmartH2O, distinguishing those that are better applicable in “small worlds” contexts, such as the Swiss case study, and those that are better suited to open, large scale scenarios, more adherent to the situation of the London and Valencia experimentations.
- **Section 7** gives an outlook on ongoing and future work.
- **Section 8** concludes with the list of references considered for the preparation of this deliverable.

1 Introduction

SmartH2O is a socio-technical system, where both users and automated algorithms jointly contribute to the achievement of water saving goals.

SmartH2O users (those of interest for this deliverable) are water consumers; they contribute to the platform in a variety of ways:

- Consuming water in buildings where consumption is recorded by smart meters and communicated automatically to the system.
- Consuming water in buildings where consumption is not recorded by smart meters and thus is communicated manually by the user to the system.
- Providing psychographic data about their households.
- Accessing educational content and water saving tips.
- Performing social actions, such as inviting friends to the platform or socially publishing/rating water saving tips and recommendations.

On the other hand, algorithms also must fulfill a variety of tasks, among which:

- They collect and verify for correctness smart meter data and manually input consumption data.
- They disaggregate consumption data to better mine water consumption patterns.
- The cluster users into classes, which exhibit homogeneous water consumption patterns.
- They compute the best water saving recommendations based on the user model.
- They collect action traces of users from multiple sources (the Consumer Portal, and the Drop! games) and implement gamification business rules that compute the most appropriate rewards in return to the users' actions.

However, SmartH2O users do not live in isolation within the walls of the platform; they have a social life outside SmartH2O, which may be partially reflected in their online activities in one or more digital social networks.

Furthermore, SmartH2O impact is not confined to the captive users of the platforms, i.e., those reached by the commercial relationships of the water utility company that provides them with the service. Also other water consumers may be targeted, in order to "spread the message" about water sustainable consumption beyond the boundaries of the SmartH2O platform.

These observations motivate the use of social networks in combination with the SmartH2O platform.

The role of social network is manifold:

- They are a vehicle for current users of the SmartH2O to share their achievements, both in the Consumer Portal and in the Drop! Game, thus acquiring "green" social status.
- They enable SmartH2O users to invite other users to try the SmartH2O platform or the Drop! Awareness games.
- The allow the collection, when a SmartH2O water consumer can be associated with a social network user, of further profile data, which can be used to enrich the user model.
- They allow water utilities customer relationship managers to identify influential users in a social network, which could be potential targets of dissemination messages and awareness campaigns.

A second, not less important, issue in a socio-technical system that makes heavy use of human input is the monitoring of the quality of users' contributions. This topic has a large tradition in computer science, and can be regarded as relevant to SmartH2O under two viewpoints:

- The detection of malicious behavior, which could spoil the SmartH2O experience for

the community of users. This may occur, for example, when water consumers overload the system with false action declarations (e.g., watching educational content, performing a water saving action outside the system) in order to gain points and climb up in the collective leaderboard or getting undue advantage in the point redemption program.

- The detection, and possibly the correction, of wrong data acquired by the users. This may occur, for example, when users are requested to manually input consumption data.

In this deliverable, we will therefore start the work on the computational exploitation of social awareness techniques, by performing an in depth survey of the fields of social network analysis, trust and influence assessment, game player behavioral analysis, and “spammed” value detection, which are fundamental for unlocking the potential of social networks, gamification, and games with a purpose for water saving.

2 Social network analysis, trust and people search techniques

Social network analysis (SNA) is a broad field, which refers to the general problem of collecting and processing data coming from the online activity of users in digital social networks, for the purpose of computing relevant properties of individual users or of communities [Otte and Rousseau, 2002].

In this section, we perform a broad review of the main contributions in the SNA field, organised according to the practical applications more relevant to the SmartH2O social awareness platform:

- We start by the general topic of graph representation of social networks, which spans the techniques for modelling, storing, mining, querying very large scale graphs, representing the structure the interrelations among users active in a social network.
- We then address the application of SNA techniques to the characterization of several properties of communities, mined from activity traces produced in social networks.
- We conclude surveying the specific problem of applying SNA techniques to the specific problem of determining influence and trust of users, which is a relevant goal for any effort aimed at using online social dynamics to forecast and/or optimize the diffusion of target messages across a vast community of users.

2.1 Social graph techniques

This section opens the survey on social network analysis by focusing on the general purpose analysis methods based on a graph representation of data, which is the most natural format for encoding the structure of communities, where nodes are users and edges represent different semantic relationships among them. The size of graphs representing real social networks poses severe performance challenges; therefore, the section also contains a survey of graph data management approaches.

2.1.1 Graph models and processing

Graphs allow universal representation of data from various application domains, such as social networks, computational biology, software models, chemical data analysis, computer networks, software bug localization, the web, to name just a few. They are capable of representing heterogeneous data and modelling complex structures and interactions within them [Bislimovska, 2014]. These graph data require efficient techniques for their managing and mining.

For example, common techniques for managing graph data are querying, indexing and storage. Furthermore, methods that analyse the graph data for discovering useful information, known as graph data mining, are also of particular interest. Graph mining is used for identification of common or useful substructures and detecting anomalous or unusual structures.

In this section, we will provide a survey of a variety of techniques for graph processing, i.e., their management and mining.

2.1.2 Graph mining

Graph mining refers to the problem of extracting interesting substructures from very large, graphs; it includes techniques and approaches for pattern mining of graphs, graph clustering, and graph classification.

Graph pattern mining is the process of discovery subgraphs from a collection of graphs or a single massive graph, with a frequency no less than a user-specified support threshold. These patterns are useful at characterizing graph sets, discriminating different groups of graphs, classifying and clustering graphs, and building graph indexes [Aggarwal et al., 2010].

SUBDUE [You et al., 2006] is a system that identifies interesting and repetitive substructures in biological networks. It is based on graph compression and the minimum description length principles. It first discovers the best graphical pattern that minimizes the description length (MDL) of itself and that of the original graph. The best pattern is included in a hierarchy and the original graph is compressed with it. This provides a basis for discovering hierarchically defined structures [Holder et al., 1994].

An iterative mining method based on partial least squares regression (PLS) is proposed in [Saigo et al., 2008]. To apply PLS to graph data, a sparse version of PLS is developed first and then it is combined with a weighted pattern mining algorithm. The mining algorithm is iteratively called with different weight vectors, creating one latent component per one mining call. The proposed method is efficient and easy to implement, because the weight vector is updated with elementary matrix calculations.

ReFeX (Recursive Feature eXtraction) [Henderson et al., 2011] is an algorithm, that recursively combines local (node-based) features with neighbourhood (egonet-based) features and outputs regional features -- capturing "behavioural" information. These regional features represent the kind of nodes to which a given node is connected (e.g., connected to rich people), as opposed to the identity of those nodes (e.g., connected to a specific person).

The regional features can be used in within-network and across-network classification and de-anonymization tasks -- without relying on homophily, or the availability of class labels.

gSpan (graph-based Substructure pattern mining) [Yan and Han, 2002] discovers frequent substructures without candidate generation and false positives pruning. gSpan builds a new lexicographic order among graphs, and maps each graph to a unique minimum DFS (depth first traversal) code as its canonical label. Based on this lexicographic order, gSpan adopts the depth-first search strategy to mine frequent connected subgraphs efficiently. It combines the growing and checking of frequent subgraphs into one procedure, thus accelerating the mining process.

Graph clustering divides a given set of objects, into groups (clusters) of similar objects. The similarity between objects is typically defined with the use of a mathematical objective function [Aggarwal et al., 2010]. Graph clustering is useful in a number of practical applications such as marketing, customer-segmentation, and data summarization.

The approach in [Cheng and Church, 2000] introduces a bi-clustering algorithm that performs clustering in two dimensions simultaneously for a given gene expression matrix of samples and genes. Statistically significant sub-matrices of a subset of genes and a subset of samples are the identified biclusters. The algorithm finds maximal sized biclusters that satisfy a certain condition on the residue scores. It uses a greedy approach that identifies each bicluster separately by iteratively removing rows and columns until the mean squared residue score for the sub-matrix is smaller than a threshold and by iteratively adding rows and columns while the quality assessment score does not exceed threshold. Each run of the algorithm identifies a sub-matrix (bi-cluster) separately, and the next bi-cluster is identified after the found sub-matrix is masked by randomization.

Molecular COmplex DEtection (MCODE) [Bader and Hogue, 2003] is an approach for identification of protein clusters that are heavily interacting. MCODE starts with weighting each node of the graph based on the density of its local neighborhood. Next, nodes with high weights are assigned as seeds and starting from these seed nodes initial clusters are obtained by iteratively including neighboring nodes to the cluster. Finally, an optional third step is proposed to filter proteins according to a connectivity criteria.

SA-Cluster [Zhou et al., 2009] is a graph clustering algorithm, based on both structural and attribute similarities through a unified distance measure. The method partitions a large graph associated with attributes into k clusters so that each cluster contains a densely connected subgraph with homogeneous attribute values. A unified neighborhood random walk distance measure is designed to measure node closeness on an attribute augmented graph. An effective method is proposed to automatically learn the degree of contributions of structural similarity and attribute similarity.

SS-KERNEL-KMEANS [Kulis et al., 2009] is a technique that optimizes a kernel-based semi-

supervised clustering objective to cluster both vector-based and graph-based data. For vector data, the kernel approach also enables to find clusters with nonlinear boundaries in the input data space. For a given input data in the form of vectors and pairwise constraints, the algorithm constructs a kernel, such that running kernel k-means results in a monotonic decrease of the semi-supervised clustering objective function at every iteration of the kernel k-means algorithm. The approach can easily be generalized to optimize a number of different semi-supervised graph clustering objectives for which constraint-based supervision is more natural.

The method in [Yan et al., 2007] is a stepwise algorithm, which constructs a neighbor association summary graph by clustering co-expression networks into groups. A neighbor association summary graph measures the association of two vertices based on their connections with their neighbors across input graphs. Once they build the neighbor association graph, they decompose it into (overlapping) dense subgraphs and then eliminate discovered dense subgraphs if their corresponding node-sets are not frequently dense enough.

A multi-level algorithm for graph clustering using stochastic flows is presented in [Satuluri and Parthasarathy, 2009]. The graph is first successively coarsened to a manageable size, and a small number of iterations of flow simulation is performed on the coarse graph. The graph is then successively refined, with flows from the previous graph used as initializations for brief flow simulations on each of the intermediate graphs. When the final refined graph is reached, the algorithm is run to convergence and the high-flow regions are clustered together, with regions without any flow forming the natural boundaries of the clusters.

In **graph classification**, there exists an assumption that the properties of interest of a certain number of graphs or a certain part of a graph are available as a training dataset, and the goal is to derive the same properties of other graphs or the remaining part of the graph [Aggarwal et al., 2010].

gBoost [Saigo et al., 2009] is a mathematical programming boosting method that progressively collects informative patterns. Boosting is a general method for improving the accuracy of any given learning algorithm. In order to apply the boosting method to graph data, a branch-and-bound pattern search algorithm is developed based on the DFS code tree. The constructed search space is reused in later iterations to minimize the computation time. The algorithm is designed such that the search space is pruned autonomously, not by external constraints. It consists of two tightly-coupled components: the machine learning part that solves the mathematical program and the graph mining part that finds optimal patterns.

Kernel methods, construct a prediction rule based on a similarity (kernel) function between two objects. The technique in [Mahé and Vert, 2009] proposes new kernels with a parameter to control the complexity of the subtrees used as features to represent the graphs. This parameter allows to smoothly interpolate between classical graph kernels based on the count of common walks, on the one hand, and kernels that emphasize the detection of large common subtrees, on the other hand. The approach also introduces two modular extensions to this formulation. The first extension increases the number of subtrees that define the feature space, and the second one removes noisy features from the graph representations.

The approach in [Kashima and Inokuchi, 2002] presents kernel methods for classification of graphs with node labels and edge labels. The kernel is used for computation of inner products for pairs of graphs represented in a feature space. More specifically, a graph kernel is designed for a pair of graphs by a random walk on a node product graph of the two graphs. The kernel represents the probability with which two label sequences generated by two synchronized random walks on the graphs are identical.

2.1.3 Graph data management

Managing graph data is a challenging task. Although graph representation of data offers greater expressive power, data representation, access and processing have higher complexity. To accomplish this task efficiently, one needs to address several important issues: how to query graph data; and how to index the data to obtain efficient query processing [Aggarwal et al., 2010].

In this section, we present approaches for searching graph-based data considering different querying and indexing strategies, as well as some graph query languages.

A classical formulation of the problem of **searching graphs** is through finding an exact or approximate correspondence between a query graph, and a data graph, known as graph matching. Applications of graph searching to social networks analysis could be, for example, the processing of queries that look for users endowed with specific types of relationships within their communities.

TALE (Tool for Approximate Subgraph Matching of Large Queries Efficiently) is a general tool for approximate subgraph matching of large graph queries studied in [Tian and Patel, 2008]. It queries graph databases and uses novel indexing method considering the neighbours of each database node, thus, capturing the local structure around each node. A database node matches a query node only if the two nodes match and their neighborhoods also match. The algorithm consists of determining important nodes in the query and probing them against the index, thus finding the best matching node pair. The degree centrality measure establishes the node importance, such that, nodes with high degrees are more important than low degree nodes. The match is expanded through the neighboring nodes of the matched nodes until no more nodes can be added to the match.

MuGram [Krishna et al., 2012] is a multi-labeled graph matching approach that handles graphs with multiple labels for both vertices and edges. It uses an indexing technique, which, beside the labels' information, contains neighborhood information for each node which allows pruning incompatible candidates at early stages of matching. The matching process is based on neighborhood connectivity check that ensures that the graph invariant property for each query node is captured by the matching reference node. An index on the query graph is maintained to avoid repeatable processing of the query graph for each query node.

The approach in [Zhong et al., 2013] explores the diversity of user information need when searching graphs differentiating between exploratory search, where the user is unfamiliar with the graph structures, and known-item search, where the user has as a target a set of trees, or particular pattern. The problem of known-item search is addressed by expressing the query as a set of keywords. The answers to the query represent minimum connected trees, that represent subtrees of an unlabelled directed weighted graph containing at least one matched node for every query keyword. Matched Vertex Pruning index is used to capture the query-independent local neighborhood information in the graph by pruning matched vertices that do not participate in the answer trees with heights less than a threshold. The approach is independent on the graph search algorithm and minimizes the index access times.

In [Lin et al., 2012], a subgraph query processing is presented that generalizes exact edge matches to path matches constrained by a path length. The order of the matching vertices is optimized by choosing the next node to be matched to minimize the search space. The work proposes three different types of indexing: distance index, that considers the distance among all pair of vertices on the graph; Frequent Pattern Index based on the Frequent Generalized Subgraph, a frequent subgraph pattern whose frequency is greater than a threshold; Star index based on a star structure in the graph where one node is chosen as central and all its incident edges to the other vertices are considered.

Lindex [Yuan and Mitra, 2013] is a lattice-structure index for efficient and fast answering of subgraph queries, reducing the subgraph-isomorphism comparisons. Each node in a lattice represents a graph, where any pair of graphs has at least upper bound and a greatest lower bound. Nodes in the index represent key-value pairs, where the key represents a subgraph in the database, and the value is a list of database graphs containing the key. An edge between two index nodes indicates that the key in the parent node is a subgraph of the key in the child node. The query answering algorithm identifies a set of maximal subgraphs in the index and obtains a candidate set of answers by intersecting direct value sets of these subgraphs. The candidate set is pruned by identifying supergraphs of the query and eliminating graphs in the database that contain these supergraphs (from the candidate set).

Graph query languages are used to manipulate graphs in their full generality. This means the ability to define constraints (graph-structural and value) on nodes and edges not in an iterative one-node-at-a-time manner but simultaneously on the entire object of interest

[Aggarwal et al., 2010].

GraphQL [He and Singh, 2008] is a graph query language in which graphs are the basic unit of information from the ground up. GraphQL uses a graph pattern as the main building block of a query. A graph pattern consists of a graph structure and a predicate on attributes of the graph. Graph pattern matching is defined by combining subgraph isomorphism and predicate evaluation. The core of GraphQL is a bulk graph algebra extended from the relational algebra in which the selection operator is generalized to graph pattern matching and a composition operator is introduced for rewriting matched graphs. In terms of expressive power, GraphQL is relationally complete.

GraphDB [Güting, 1994] is an object-oriented data model and query language for graphs. In the GraphDB data model, the whole database is viewed as a single graph. Objects in the database are strong-typed and the object types support inheritance. Each object is associated with an object type and an object identity. The object can have data attributes or reference attributes to other objects. There are three kinds of object classes: simple classes, linked classes, and path classes. Objects of simple classes are nodes of the graph. Objects of link classes are edges and have two additional references to source and target simple objects. Objects of path classes have a list of references to node and edge objects in the graph. A query consists of several steps, each of which creates or manipulates a uniform sequence of objects, a heterogeneous sequence of objects, a single object, or a value of a data type. The uniform sequence of objects has a common tuple type, whereas the heterogeneous sequence may belong to different object classes and tuple types. Queries are constructed in four fundamental ways: derive, rewrite, union, and custom graph operations.

2.1.4 Evaluation of Social Graph Techniques

In this section we provide an evaluation of the techniques for mining and management of graph data.

In Table 1 we present the general graph mining techniques with respect to the following criteria:

- **Simplicity of implementation:** Out of all graph mining methods, graph classification methods based on graph kernels are the simplest to implement. However, this comes at a price of higher complexity, which is especially important when the graph/graphs is/are large.
- **Scalability:** Increasing the workload strongly affects graph mining approaches. For example, in graph classification and pattern mining, enumerating all the graphs/ subgraphs even when graphs are not large is computationally expensive. To tackle this problem, methods that mine the most frequent/significant patterns and subgraphs have been developed.
- **Examination of entire graphs/graph nodes within a graph**: Methods for graph mining differ based on whether they require examination of entire graphs or the nodes within a graph which depends on the context of the particular application and method.

Table 1. Evaluation of Graph Mining Techniques.

	Scalability	Simplicity of implementation	Examination of Entire Graph	Examination of Graph Nodes
Graph Clustering	X		X	X
Graph Classification	X	X	X	X
Graph Pattern Mining	X		X	

In Table 2 we illustrate an evaluation of the graph data management techniques, i.e., graph search and graph query languages considering the following criteria:

- **Sensitivity to intermediate representation of data**: Indexing applied for graph search is particularly sensitive to the way graph data are represented; Change of graph presentation, for example, considering more or less details, might require change of the corresponding indexes.
- **Query expressiveness:** high query expressiveness is achieved by using graph query languages represented by a set of well defined constraints. This allows manipulation with graphs in their full generality. Although graph search techniques are able to handle complicated queries, users, in general, have a vague notion of what they query for, so lack of constraints might cause the incorrect query formulation.
- **Scalability**: Existing graph search algorithms have difficulties in being parallelized in order to become scalable for larger graph data sets. Graph query languages are more robust to parallelization.
- **Inexact matching**: A variety of graph search techniques are able to tolerate inexact matching among queries and graphs. Existing approaches for graph query languages deal with exact matches.

Table 2. Evaluation of Graph Data Management Techniques.

	Sensitivity to intermediate representation	Query Expressiveness	Inexact matching	Scalability
Graph Search	X		X	X
Graph Query Languages		X		X

2.2 Community Detection and Roles Analysis

SNA and graph based techniques can be used to investigate individual and collective properties, mined from social network activity traces. In this section, we review two problems that are directly related to the goals of SmartH2O:

- Analysing social network data in order to discover the presence of subsets of users with distinguished properties, which characterize them as members of a community.
- Within a community, analysing the online activity traces of users in order to classify them in terms of their behaviour.

2.2.1 Community Detection

There is no universally accepted definition of community, but it often depends on the specific system and application that are considered [Fortunato, 2010]. In the context of a social network, which is typically represented as a graph, a community can be vaguely defined as **a group of nodes more densely connected to each other than to nodes outside the group** [Java et al., 2007]. Communities can be also seen as what [Wasserman and Faust, 1994] described as cohesive subgroups, namely subsets of actors with relatively strong, direct, intense, frequent or positive ties.

Communities can be distinguished as implicit and explicit [Papadopoulos et al., 2011; Zafarani et al., 2014]. An explicit community is created as a result of human decision and acquires members based on human consent. In social media, examples of explicit communities are groups in Facebook or LinkedIn, where members join consciously. Implicit communities, on the other hand, consist of people who have something in common. These groups are assumed to exist in the system and are waiting for being discovered [Papadopoulos et al., 2011] but they and their members are usually obscure to many people [Zafarani et al., 2014]. Users that follow Twitter accounts relevant to water sustainability, like

SmartH2O, comprise such an implicit community because of their common interest.

Community discovery can be viewed as a data mining analysis on graphs, i.e. an unsupervised classification of its nodes. This is the most studied data mining application on social networks. Other applications, such as graph mining (see section 2.1.2), are still in an early phase of their development. Community discovery, however, has achieved a more advanced development with contributions from different fields [Coscia et al., 2011].

Community detection algorithms are often provided with a graph, where the vertices represent individuals (e.g. users) and the edges stand for a relationship between the individuals (e.g. friendship). Thus, in such a graph, community detection addresses the identification of groups of vertices that are more densely connected to each other than to the rest of the network.

A novel approach to community discovery is often designed for a specific problem and thus develops its own definition of community [Coscia et al., 2011]. Furthermore, communities have a number of interesting features, e.g., they can exhibit a hierarchical or overlapping configuration of the groups inside the network, and the graph can include directed or multi-relational edges. Consequently, there is diversity in definitions and features that led to a rich literature on the community detection problem. Yet, most overviews in literature tend to focus on the operational method, i.e. how communities are detected from the inferred graph [Coscia et al., 2011]. In our work, we attempt to address both the operational part of the graph community detection (see Section 2.2.1.1) and the entire methodology of the studies, namely how the real world was simulated in the graph and what the underlying definition of community was (see Section 2.2.1.2).

Importance and advantages of Community Detection

Community structure can reveal rich hidden information about complex networks that is not easy to detect by simple observation [Liu et al., 2014]. The advantages of community detection depend primarily on the context of the application and the definition of community but some are commonly intended.

Identifying communities and their boundaries provides the opportunity to classify the vertices according to their structural position in the modules. Vertices with a central position in their clusters may have an important function of control and stability within the group. Similarly, vertices between different modules play an important role of mediation and are critical for the relationships and exchanges between different communities [Fortunato, 2010]. Thus, group leaders can become visible as well as key group connectors [Coscia et al., 2011].

Communities can be further analysed in terms of content or other attributes, to get further insights into community interests or other information that may indicate what constitutes each particular community. Moreover, communities can be described by examining only the their central users like in the work of [Gupta et al., 2012].

Especially in the case of Twitter, detecting and analysing communities and roles of particular importance can help towards the diffusion of information. E.g., contacting or passing information to users that are mediators between communities increases the likelihood of further spreading information to the communities they are attached to (more about community roles in Section 2.2.2). In addition, the content of the tweets within a community or user profile properties, e.g. their location, can be analysed.

2.2.1.1 Generic Graph-based Methods for Community Detection

Community Detection in SNA graphs is a very extensive and active research topic with a rich literature of methods, algorithms for community identification and evaluation methods. An extensive overview of graph community detection techniques is provided by [Fortunato, 2010]. A review of methods particularly in the context of Social Media [Papadopoulos et al., 2011] surveys methods also in terms of computational complexity and memory requirements. Community detection and graph clustering methods can be classified, based on their methodological principles, into five classes:

Subgraph Discovery, (like CPM [Palla et al., 2005]) methods that are based on presumed specification of the structural properties that a subgraph of the network should satisfy in order

to be considered a community.

Vertex Clustering, methods that originate from the traditional data clustering research and typically cast a graph vertex clustering problem to one that can be solved by conventional data clustering.

Community quality optimization, methods that are founded on the basis of optimizing some graph-based measure of community quality.

Divisive, methods that are based on identifying the network elements (edges and vertices) that are positioned between communities (and typically remove them progressively).

Model-based, (e.g. Infomap [Rosvall and Bergstrom, 2008] and Label Propagation methods like SLPA [Xie and Szymanski, 2012; Xie et al., 2011]) are a broad category of methods that either consider a dynamic process that happens on the network, and reveals its communities, or they consider an underlying model of statistical nature that can generate the division of the network into communities.

Most methods handle the problem of community detection by trying to identify non-intersecting subgraphs of nodes where the detected communities are considered mutually exclusive sets of nodes and, thus, a node only belongs to one community.

However, in real network cases, **communities often overlap** and one person can have different roles in different communities [Java et al., 2007]. Community overlap is important for Social Media networks since it is common for entities to participate in multiple communities. For instance, a user may be affiliated to his/her family, friends and professional community [Papadopoulos et al., 2011]. In the case of Twitter, a user may be an information seeker in one community but an information source in another. This is inline with concept according to which a user may have different topics of interests as a writer and different as a reader [Welch et al., 2011] and thus be involved in two or more communities with different roles. Furthermore, special roles like bridging users and bridging connections can be investigated with algorithms capable of detecting overlapping communities and assigning nodes to more than one group [Grabowicz et al., 2012].

The detection of overlapping communities is a challenging problem with increasing interest in recent years because of the natural attitude of individuals in real-world networks to participate in multiple groups at the same time [Amelio and Pizzuti, 2014]. There is already a great volume of community detection algorithms for overlapping communities and consequently some good overviews on this specific type of methods have been provided. One review of the main proposals in the field describes methods for static networks but also some new approaches that deal also with dynamic networks that change over time [Amelio and Pizzuti, 2014]. An extensive review of the state of the art of overlapping community detection algorithms, quality measures, and benchmarks provides a thorough comparison of 14 different algorithms, resulting in a guide for applications in this field [Xie et al., 2013].

Therefore, in the context of SmartH2O, we consider the presence of overlapping communities in Twitter and examine the following methods that share the same view and are most relevant.

CPM (Clique Percolation Method) [Palla et al., 2005] is one of the oldest methods developed for overlapping community detection. The method is based on the assumption that a community consists of overlapping sets of fully connected subgraphs. It detects communities by searching for adjacent cliques [Xie et al., 2013] and adopts the observation that a typical member in a community is linked to many other members, but not necessarily to all other nodes in the same community [Java et al., 2007]. It starts by initially identifying all cliques of size k in a network. Once these are identified, a new graph is constructed in a way that vertexes represent these k-cliques. Two nodes are connected if the k- cliques that represent them share k−1 members. Connected components in the new graph identify which cliques compose the communities and since a vertex can belong to multiple k-cliques simultaneously, overlap between communities is possible [Xie et al., 2013]. CFinder[1] is the implementation of CPM while CPMw was later introduced [Farkas et al., 2007] for weighted

[1] Available at www.cfinder.org

networks. According to [Fortunato, 2010], CPM assumes that the graph has a large number of cliques, so it may fail to give meaningful covers for graphs with just a few cliques, like technological networks and some social networks. On the other hand, if there are many cliques, the method may deliver a trivial community structure, like a cover consisting of the whole graph as a single cluster.

SLPA [Xie and Szymanski, 2012; Xie et al., 2011] is a general speaker-listener based information propagation process [Xie et al., 2013]. Each node is endowed with a memory to store received labels and can have both the role of listener and speaker. In the case of a listener, it takes labels from the neighbors and accepts only one following a listening rule, e.g. the most popular observed at the current step. If it is a speaker, it sends a label to the neighboring listener node by choosing a label with respect to a certain speaker rule, such as singling out a label with a probability proportional to its frequency in the memory. The algorithm ends when a predetermined number of iterations has been reached [Amelio and Pizzuti, 2014]. The number of communities is not required in SLPA. Rather, it is determined by the clustering of labels in the network. Furthermore, it can also be adapted for weighted and directed networks by generalizing the interaction rules, known as SLPAw [Xie et al., 2013].

Sequential Clique Percolation algorithm (**SCP**) [Kumpula et al., 2008] is a fast implementation of CPM [Fortunato, 2010]. It detects k-clique communities by sequentially inserting the edges of the subject graph, one by one, starting from an initial empty graph. It finds clique communities of a given size [Xie et al., 2013]. According to [Fortunato, 2010], its biggest advantage is the implementation for weighted graphs while it is faster than the original implementation of CPM [Fortunato, 2010; Xie et al., 2013].

OSLOM [Lancichinetti et al., 2011] is a widely employed method in the area of community detection. OSLOM is based on a topological approach to detect statistically significant clusters. First, a null model of graphs obtained by reshuffling the connections of the given network is considered and then the probability of finding each group in the ensemble formed by these random graphs is estimated. Under the assumption that an optimized clustering technique has been applied to the random graph, techniques from the statistics are necessary to properly evaluate the probability of each group. The method incorporates a local search method for the exploration of the network with the aim of finding clusters that improve the estimated probability, namely to find groups that have lower probability of existence in random graphs. OSLOM provides a set of clusters at the lowest hierarchical level, a list of nodes belonging to several groups and those not belonging to any group [Grabowicz et al., 2012]. Its popularity can be also attributed to the fact that it presents many advantages over other methods. Apart from its ability to detect overlapping communities, it can handle both directed and weighted networks. Furthermore, it is suitable for dynamic networks and it accounts for singleton communities, i.e. communities of one node.

MOSES [McDaid and Hurley, 2010] greedily expands a community from edges [Xie et al., 2013], optimizing a global objective function based on a statistical network model. In the model of this method, a graph is represented as a random symmetric adjacency matrix and the objective function computes the maximum likelihood estimators from the observed likelihood [Amelio and Pizzuti, 2014].

The approach in [**Yang and Leskovec, 2015**] presents an evaluation methodology for comparing network community detection algorithms based on their accuracy on real data, under assumption that the goal of network community detection is to extract functional communities based on the connectivity structure of the nodes in the network. The approach identifies networks with explicitly labeled functional communities, referred as ground-truth communities. Another branch of this work studies the problem of community detection from a single seed node. After performing comparison with the class of scalable parameter-free community detection methods, it was shown that the proposed methods reliably detect ground-truth communities.

An algorithm for optimizing modularity that allows one to study networks of unprecedented size is introduced in [**Blondel et al., 2008**]. It is a heuristic method that is based on modularity optimization. Its accuracy has been tested on ad-hoc modular networks and it is

shown to be excellent in comparison with other (much slower) community detection methods. By construction, the algorithm unfolds a complete hierarchical community structure for the network, each level of the hierarchy being given by the intermediate partitions found at each pass. The limitation of the method for the experiments is the storage of the network in main memory rather than the computation time.

The approach in [**Leskovec et al., 2010**] explores a range of network community detection methods in order to compare them and to understand their relative performance and the systematic biases in the clusters they identify. To achieve that, it evaluates several common objective functions that are used to formalize the notion of a network community, and examines several different classes of approximation algorithms that aim to optimize such objective functions. Considering community quality as a function of its size provides a much finer lens with which to examine community detection algorithms, since objective functions and approximation algorithms often have non-obvious size-dependent behavior. The aim is not to find the "best" community detection method or the most "realistic" formalization of a network community, but, to understand the structural properties of clusters identified by various methods, and then depending on the particular application one could choose the most suitable clustering method.

2.2.1.2 Holistic Community Detection Approaches

In section 2.2.1.1 we referred to methods that are able to identify communities in a graph, without examining how the input graph occurs and how the community is defined. In this section, we will review methods holistically, i.e. we will examine how approaches defined communities and represented real-world networks.

In the last years there has been strong interest in the literature to develop methods and algorithms that can efficiently highlight this hidden community structure of real networks. Traditionally, this is achieved by partitioning the graph. Network representation, however, can be very complex with different variants in the graph model and, thus, the various approaches in the literature focus on some of these properties and establishing their own definition of community. Then, according to this definition the methods extract the communities that are able to reflect only some of the features of real communities [Coscia et al., 2011].

Thus the approaches in literature present a large diversity on several points, and it is difficult to classify them in a scheme of exclusive categories. They mostly present differences in the information they exploit as an input (topology, linguistic information, interactions, common attributes etc.), in how they represent the real-world network, and in the definition of a community. They are also differentiated by whether or not they consider important features of a graph network like directionality and weighted edges, and whether the creators consider the existence of overlap among communities.

[Coscia et al., 2011] classifies community definitions into **Density-based** communities that are defined entirely based on the topology of the network, **Vertex Similarity-based**, which comes from the assumption that communities can be groups of nodes that are similar to each other with respect to some reference property and irrespectively of whether or not they are connected by an edge, **Action-based** where nodes are grouped based on actions they perform (also irrespectively to whether they are directly linked), and **Influence Propagation-based** where nodes are grouped when they perform same actions as an influence of their leaders.

With respect to type of connections, [Ding, 2011] distinguishes between social connection and the similarity connection. Social connections are the real connections that appear in the networks: a friendship, a co-authorship, a communication between people, etc. A similarity connection instead does not physically exist but is derived. Such similarity connections are usually represented with quantified similarity metrics, for example the number of common hashtags two users used in Twitter, upon the assumption that the more hashtags they use in common the higher the probability of similar interests. Therefore, as an alternative, methods could be respectively categorized into the ones that use actual social connections, the ones that are based in similarity connections and the methods that integrate both connection types.

Literature approaches can be thereby categorized in different schemas, and fall in categories

often non-mutually exclusive. To present the different approaches of the literature with focus on Social Media and especially Twitter. We present the reviewed methods categorised by the considered type of connection between users.

Similarity-based community detection methods

One can assume that communities are groups of vertices similar to each other, where the similarity between each pair of vertices can be computed with respect to some reference property, regardless of whether they are actually connected by an edge or not [Fortunato, 2010]. Therefore, we classify methods, in which users are grouped in communities based on common attributes they present, as similarity-based methods. In Twitter and social networks the individuals may present various types of similarities (e.g. users who like same pages in Facebook etc.). These similarity attributes can be heterogeneous, like a combination of common used words, common friends and location. In this case, we usually talk about communities of common interests.

In [Zhang et al., 2012], the authors constructed graphs in which the nodes represent Twitter users, and the edges are derived from a calculated similarity based on textual features (text content, URLs and hashtags) and social features (follow and retweet relationships). Regarding the latter, the "following"-similarity is based on common friends and followers between two users, while the "retweeting"-similarity is derived from a combination of the fraction of people they commonly retweet and the frequency they retweet each other. An aggregated similarity distance was defined and then a typical k-means clustering algorithm was applied. This is from the few studies that didn't actually infer a graph but handled the problem as a normal unsupervised clustering task. Consequently, they detected clusters of users of common interests that were non-intersecting.

The study of [Beguerisse-Díaz et al., 2014] on influential Twitter users during the 2011 riots in England differed mostly in the intentions for community detection. Directed Markov Stability was applied to detect first interest communities that reveal user groupings according to location, profession, employer and topic and secondly, communities of similar flow-roles, which resulted in a classification of users into five flow-based roles. For the second part, they calculated pairwise similarities between the users in terms of incoming and outgoing directed paths and inferring a respective graph, on which they applied community detection.

In [Greene et al., 2012] the feature of lists provided by Twitter and the fact that there exist already several lists curated by users were exploited, bringing an interesting perspective of utilising the user categorization that has been made manually by other Twitter users. In this approach, a graph was constructed where the nodes represent Twitter lists and are connected with weighted edges based on how many users they share. These weights are a measure of similarity between two lists. The aim is to create communities of Twitter lists, which can be also overlapping and to use the metadata of the lists to label and evaluate the resulting communities. For the graph community detection process, they applied OSLOM to identify communities of overlapping user lists but in combination with a consensus clustering technique, the aim of which was to generate and combine the result of different overlapping community sets in order to produce more stable results.

Topology-based community detection methods

This group of methods includes methods that are based on **topology** because they take into account the graph structure of the actual network, using the stated connections like friendships or follow-links in the case of Twitter.

In the [Java et al., 2007] study, a directed graph was used where the nodes, representing Twitter users, were linked by the typical friend relationship links (follow) and overlapping communities were detected with the Clique Percolation Method (CPM) and further analysed.

In the [Lim and Datta, 2012a, 2012b] approaches, celebrities in Twitter were used as reference points and representatives of certain categories (topics) and then communities were detected based on friendship links (i.e. reciprocal stated friendships) among their followers, on an undirected graph. A similar approach was followed later in [Lim and Datta, 2013]. The use of topological follow links was based on the authors' assumption that they are easy to collect and therefore an advantage for the community detection methods. This idea

might present potential for communities of common interests but requires nonetheless some manual semantic annotation of the celebrities used as seeds. The Clique Percolation Method (CPM) and the Infomap [Rosvall and Bergstrom, 2008] algorithm were applied for the detection of the communities on the graph.

In [Grabowicz et al., 2012] the typical follow links were exploited to detect overlapping communities on a directed graph and examine the different types of interaction of mention and retweet in respect to their position in the detected communities (within a group, between groups, intermediary or to no-groups). They applied OSLOM for the community detection process due to the size, density, and directness of the network and in order to capture the possible inclusion of users in multiple groups or in none. However they performed also the analysis with a few other popular algorithms, including Infomap, Moses and Louvain. Last, in order to use some of them (Moses, Louvain), the network was symmetrized and directionality of the links was ignored.

Interaction-based community detection methods

Methods in the following group were built based on interactions between users in Twitter, therefore we characterize them as interaction-based.

In [Correa et al., 2012] an algorithm (iTop) to detect interaction-based topic-centric communities in Twitter was developed. It incorporates multiple interactions like retweets, replies and mentions, forming thus a directed weighted graph, where each node signifies a Twitter user and edge weights are derived from the frequency of their interaction, and are therefore proportional to the extent of interactivity. However, it is worth mentioning that their model starts from a topic as input and based on this creates the interaction-based graph and detects the communities. To detect communities, the LM [Clauset, 2005] algorithm for directed graphs was used.

In [Lim and Datta, 2012c] the celebrity-seed concept was explored again. Their main objective was to identify communities of users that share common interests but also communicate frequently about their common interests (HICD). The approach was compared to a previous topological approach (CICD) [Lim and Datta, 2012a, 2012b], based on follow-links. The approach uses the frequency of direct tweets between users to construct a network of weighted links and then constructs a new one keeping only the edges that exceeds a pre-determined threshold resulting in a directed graph. CPM and Infomap were applied for the detection of the communities in each set of users that follow celebrities of a specific category.

The [O'Callaghan et al., 2013] study examined communities and inter-community relations between identified Twitter accounts of 8 countries involved with extreme right groups. The inferred graph was an interactions network of nodes (accounts) connected by weighted undirected edges, which represented only the reciprocal interactions (mentions and retweets) with weights corresponding to the frequency of interaction. In this work consensus communities were used and detected applying OSLOM. Despite the fact that the method supports directionality, only reciprocal interactions were used to capture stronger relationships.

The last abovementioned methods were based on actual interaction that took place between users in Twitter aiming at identifying interactive communities. However all three works intended to discover interactive communities that occur as well from a specific topic or interest which is pre-selected, or, in general, communities of people that interact heavily on common interests.

Hybrid community detection methods

Approaches that combine different of the above aspects are classified as hybrid, since they exploit both structural and similarity-derived information

In the study of [Gupta et al., 2012] the aim was to identify communities of similar people and to describe the communities by their top central users. A similarity metric was defined as a sum of content similarity (number of common words, hashtags and URLs), link similarity (how often two users link with each other or a third user) and similarity of their location (called Meta-data similarity). The used dataset contained tweets of 3 crisis events and Spectral Clustering was performed for the community detection. It is mostly a similarity-based

methodology but incorporates as well some actual social connections between users and is thus listed here.

In the [Deitrick and Hu, 2013] approach, the community detection begins from the standard directed friend-follower network, but then supplementary features are integrated to the graph on a daily basis, by incrementing edges weights and then community detection takes place again. The supplementary features were the three Twitter interaction links, i.e. reply, retweet, and mention, and additionally the hashtag and sentiment classification (the edge weight between two users would be increased every time the used the same hashtags with the same sentiment). This study aimed at communities of users who are connected, interact and present relevant similarities in topic and sentiment at the same time. Community Detection was performed with the application of both Infomap and SLPA algorithms, which were chosen because they are able to handle both weighted and directed networks, execute relatively quickly on large graphs, and because their operation differs greatly [Deitrick and Hu, 2013], with best results achieved using Infomap.

2.2.2 Community Roles Analysis

Community Detection often aims at the revelation of community roles or is combined with it for a more insightful analysis of a network.

The identification of community roles and the individuals who hold them can be beneficial in identifying key individuals for information diffusion. For example, intermediary users belong to multiple groups and play an important role in the spreading of information because they acquire information from one group and launch messages targeting the other groups of which they are members. At the same time, the access to new information can transform them into attractive targets to be retweeted by their followers [Grabowicz et al., 2012]. So, their identification can be highly beneficial in the spread of a campaign, like a social awareness campaign in the SmartH2O context.

There is no universal definition of a role, even if it is intuitive as a concept. Furthermore, defining a social role depends on the analysis context [Forestier et al., 2012]. Borrowing the definition of [Welser et al., 2007] a role comprises a combination of particular sets of behavioural, meaningful and structural attributes. A role is highly associated with specific social interactions and relations. Using the authors' example, the social role of a father is associated with and identified by a specific set of interactions, expectations and social relations.

Social roles are often inherently defined in relational terms, namely a role only exists in relation to others who are likewise enacting social roles [Gleave and Welser, 2009].

Actors with similar roles will share common features and common patterns of relations even if they do not share any direct relationship [Forestier et al., 2012] and therefore people with similar roles share common features and communication patterns.

In the context of online social networks and communities, the concept of role is naturally present. Moreover, data derived from online settings are ideal for studying roles because they allow researchers to simultaneously bring network structure, behavioral patterns, and the meaning of interactions (via content analysis) to bear on the task of accurately identifying roles [Welser et al., 2007]. In the literature approaches in this field, roles may be pre-defined, which is also the most common case (e.g. in the approaches [Golder and Donath, 2004],[Tinati et al., 2012]) or they may emerge through observation of patterns of interaction or behavior (e.g. in [Beguerisse-Díaz et al., 2014]).

According to [Nolker and Zhou, 2005] the roles within a community can be defined by relationships between different members, member behaviours, or a combination of the two. They distinguished the roles within a community into ***relationship-based roles***, i.e. the ones that follow the traditional network structure (e.g. brokers) and ***behavior-based roles*** (e.g. debaters), i.e. those that are defined based on behavioral patterns, and state that the identification of roles is more of a point of reference than a factual statement.

According to [Gleave and Welser, 2009], social role analysis has developed two general methodological approaches: ***interpretive*** and ***structural***. Interpretive analyses employ

methods such as ethnography, content analysis, and surveys to capture behaviors and relations within groups. Structural analyses, on the other hand, employ social network analysis to differentiate individual nodes in a network based on metrics of structure drawn from data on networks. Typical roles that occur from such analyses are hubs, brokers and bridges [Denning, 2004].

However, both methodological approaches seem to present some weaknesses. Interpretive studies often neglect the macro social structure in which these roles exist and this results in role definitions that are difficult to compare across social settings. Structural analyses on the other hand, despite their contribution of important knowledge about the structure of a network and the possibility to use metrics, have moved away from the context and content of the relations. This becomes obvious with the example of the broker role, i.e. a person who links two otherwise disconnected groups of people. Although this concept is theoretically important, it does not effectively distinguish individuals who hold this position and engage in brokering behavior from the ones who do not [Gleave and Welser, 2009].

Therefore [Gleave and Welser, 2009] suggest that integrating the two approaches by combining the pure structural approach of social network analysis with the behavioral notion of a role would makes it possible to identify the social roles that are meaningful at the interaction level.

Based on this categorization of [Nolker and Zhou, 2005] into relationship-based and behavior-based roles, we present in the following subsection some relevant studies of community roles analysis in social media with focus on Twitter.

2.2.2.1 Relationship-based roles

Relationship-based or structural roles derive from social network analysis. Some of these studies have defined the roles concretely with respect to the connectivity of the nodes to the known or detected communities, namely based on their intra- and inter-community positions.

A frequently used model of community roles upon which many approaches are built has been introduced by [Guimerà and Nunes Amaral, 2005]. Their 7-role scheme was based on the structural position of the nodes with respect to detected modules (communities) and, particularly, based on their patterns of intra- and inter-module connections. The non-hub roles were:

- ***ultra-peripheral nodes***, the nodes that are linked only within their module,
- ***peripheral nodes***, which have links are within their module,
- ***non-hub connector nodes***, which present many links to other modules, and
- ***non-hub kinless nodes***, which have links homogeneously distributed among all modules.

Similarly, the hub nodes can be divided into three different roles:

- ***provincial hubs***, which are mostly connected within their module,
- ***connector hubs***, which present many links to most of the other modules, and
- ***kinless hubs***, which have links homogeneously distributed among all modules.

In the work of [Scripps et al., 2007], 4 community-based roles were introduced, defined according to the number of communities a node is connected and its degree:

- ***Ambassadors***, highly connected nodes to different communities,
- ***Big Fish***, very important (high degree) but within a community,
- ***Bridges***, low degree but connected to several communities, and
- ***Loners***, who present both low degree and community score.

Last, in [Fagnan et al., 2014], a study of the dynamic change of the roles of the individuals under the appearance of certain events, the intention was to define roles under the two role scopes; the roles would be either global to the community or limited to the community while some roles may have both global and community bound versions. Furthermore, these roles should reflect on any network, although the names may not apply exactly. The following roles

for social networks were defined:

- ***Outlier role**,* held by individuals that do not participate in any community,
- ***Principal role**,* highly connected individuals further divided into:
    - ***Community leaders***, as highly connected within a the induced community they belong to, and
    - ***Global principals**,* the ones with high centrality within the entire network,
- ***Peripheral role**,* individuals with the lowest connectivity in both their community and entire network,
- ***Mediator role***, central individuals, who don't belong solely to one community and depends on the mining algorithm they belong either to multiple communities or they are excluded from a community but are still highly connected to more than one communities, and
- ***Extrovert role***, the community equivalent of mediator, i.e. nodes that belong to a community but have more inter-community than intra-community edges.

Other structural approaches do not focus on concrete detected communities but infer the roles using typically social network analysis measures.

In [Tyshchuk et al., 2013] the authors focused only on leaders and their role in Twitter during emergency cases and defined 3 types of leaders:

- ***Diffuser***, the leader who diffuses the information through the network (associated with OutDegree centrality),
- ***Gatekeeper***, a node that controls an information flow in the network, distinguished further into critical gatekeeper (high betweenness centrality, low power) and unique access gatekeeper (low betweenness & high power values), and
- ***Information broker***, who has access to valuable information and brokers it to other nodes upon request (high inDegree centrality and power measures).

In the work of [Beguerisse-Díaz et al., 2014] a different approach was followed. Studying a sample of active Twitter users during the UK riots in 2011, they aimed at detecting information flow-based roles and at going beyond the very typical and broad leader/follower or source/sink dichotomy. Thus, they inferred a graph where users, represented as nodes, were connected with each other based on how similar flow-roles they hold in the original network. Their method concludes in a role-based similarity matrix, which provides a grouping of nodes according to their function in information propagation. Then they applied a graph-theoretical community detection algorithm to the respective graph and reveal 5 groups of nodes, i.e. communities of users, with similar in- and out-flow patterns. Some of the identified groups corresponded indeed to the traditional roles of listeners/followers and leaders, but there was also a distinction between different types of leaders, followers and intermediate roles. The flow-based roles were:

- ***References***, typically institutional accounts, important sources of content, or popular personalities with large audience of followers, who themselves follow only few accounts,
- ***Engaged leaders***, institutional and personal accounts with large number of followers who also follow other users and often interact with the public,
- ***Mediators***, users who interact with the two leader categories, as well as with nodes in the listener categories below, many of which were found to belong to journalists and reporters,
- ***Diversified listeners***, accounts with few followers that follow many nodes from different categories and suggest diversity in their interests and sources of information, and
- ***Listeners***, accounts with few followers (within the particular used network) who follow mostly Reference nodes and can be considered as passive recipients of mainstream content in this particular network.

2.2.2.2 Behaviour-based roles

Part of the difficulty in defining concretely the concept of role becomes obvious in the case of roles identified based on behavioural patterns. A segmentation of users based on their behaviour leads often to literature works that talk about user types instead of roles. For example, in [Brandtzaeg and Heim, 2011] the 'Lurker' is part of a user typology, while in [Golder and Donath, 2004] it is a considered a role. However as discussed earlier, a role is by definition a concept associated also with behavioural attributes while a media platform can be seen as a large community where users are related to each other and interact. Furthermore, behavioural patterns entail in many cases information about the interactions. Measuring, for example, the retweet ratio of a user still conveys information about the user's relationships and interactions within the broader community of Twitter.

In the following, we refer to studies on behavioural patterns that are most relevant to us.

In [Welch et al., 2011], though authors' purpose was not to define or reveal a schema of roles, they point that in Twitter every user holds dual roles, as a writer or author of his own posts and as a reader or subscriber to other's posts and that in fact these two roles not necessarily present same topical interest and that what a user wants to read may differ from what the user wants to write or share. Between these two scenarios, they mention a third one about users who mostly repost content of others acting primarily as a filter to the content of their friends.

In [Java et al., 2007] a taxonomy of users was built based on their intentions for using Twitter. Based on the content of the messages they distinguish 4 classes of users:

- ***Daily Chatters***, who share information about their daily routine,
- ***Conversations***, when user addressed others with replies or mentioning,
- ***Sharing information***/URLs, and
- ***Reporting news***, user who report or comment on current events.

while in terms of links and network information, they concluded in the following 3 categories:

- ***Information sources***, the users who post frequently and have a large audience,
- ***Friends***, and
- ***Information seekers*** follow regularly people but post seldom.

[Tinati et al., 2012] categorized users in Twitter by specific roles based on their communication behaviour, aiming mostly at identifying users who are potentially producers or distributers of valuable content. They defined 5 not mutually exclusive roles:

- ***Idea Starters***, highly engaged users, who utilize multiple social media sources, have a limited network of high quality connections. They may start the idea or just create a fertile environment for it.
- ***Amplifier***s, users who collate multiple thoughts and opinions and spread the knowledge to their large network of connections, within which they are trusted.
- ***Curators***, take the ideas of the two aforementioned and validate, question or dismiss them. They are connected to a large audience, and often select information outside their primary community of interest and tailor it to their circles of interest.
- ***Commentator***, users who contribute their own insights to the conversation, but without becoming too immersed in it and without seeking recognition. They participate in something they strongly feel about.
- ***Viewers***, users who present a passive interest in the conversation, they consume information without sharing it online, may share it though with their offline network.

To identify the roles they constructed metrics for each role, employing mostly the rank of user in cascades of information (ordered list of tweets), and in the case of transmitters, whether they belong to a different community than the users that transmitted the information before him.

In [Maulana and Tjen, 2013] communities in Facebook and Twitter were examined but with an e-word-of-mouth perspective. The following 6 clusters of users were identified:

- ***Angels***, individuals interested for business networking & sharing knowledge and advice,
- ***Active Learners***, users who learn actively and share their knowledge,
- ***Passive Learners***, users who gain knowledge from social media but rarely share their own,
- ***Social Networkers***, motivated to extend their networks, often attracted by discussions,
- ***Journalist-Narcissists***, individuals who enjoy sharing their accomplishments, seek for attention, share regularly pictures and have a general focus towards themselves, and
- ***Screamers***, impulsive users who tend to share everything that they encounter, providing often a meaningless content to the community.

While Twitter is primarily in focus, a couple of other works on role detection in other platforms are worth mentioning. Usenet is a bulletin-board service on which extensive research has been made in the context of roles and user types. It is much different than Twitter but it is worth mentioning the role schema of [Golder and Donath, 2004] because it still presents similarities and, as the authors suggest, some roles (like the 'Newbie') can be found in every virtual community. The taxonomy of social roles they constructed comprises the following roles:

- ***Celebrity***, a central figure describing the users who contribute actively to the community and are popular through their frequent activity, often defining significantly what the community is,
- ***Newbie***, the new users who have little communicative competence and often also little common ground with the group,
- ***Lurker****,* describing users who read the conversations but do not participate, and the unhealthy for the community roles of:
- ***Flamer***, users who intimidate others through aggressive language and controversial speech
- ***Ranter***, describing people seeking pointless debates, and
- ***Troll***, describing these users who deceive initially others pretending to be something or someone else to use it later as leverage for intimidation.

It becomes clear that the three main roles of Celebrity, Newbie and Lurker could be characterised as universal since they can fit the context of several online communities, including Twitter. The Lurker, for example, is often used in research either as a role, like here, or as a user type [Brandtzaeg and Heim, 2011] in order to describe those users who don't appear to participate a lot in the community and therefore it is difficult to interpret their behaviour because of general lack of traces. Thus it can be applied to different types of electronic communities. Similarly, behaviours associated with the unhealthy roles above appear as well in Twitter and other platforms, along with other malicious behaviours (e.g. spamming). Thus, despite the differences in the social media platforms, it is likely that some roles apply to different platforms of course with an equivalent that fits the specific online community.

Last, another study on Usenet came from [Nolker and Zhou, 2005] who identified the two key roles of **Leaders** and **Motivators**, and as non-supportive roles the **Chatters**.

Table 3 summarises the above studies in a short overview. The studies that were based on specific social media platforms are designated.

Table 3. Community role schemas in literature.

Relationship-based	**Behavior-based**

Nodes: *ultra-peripheral* *peripheral* *non-hub connector* *non-hub kinless* Hubs: *provincial* *connector* *kinless*	[Guimerà and Nunes Amaral, 2005]	Intention-based: *Daily Chatters* *Conversations* *Sharing information/URLs* *Reporting news* Information-based: *Information sources* *Friends* *Information seekers*	[Java et al., 2007] *in Twitter*
Ambassadors *Big Fish* *Bridges* *Loners*	[Scripps et al., 2007]	*Idea Starters* *Amplifiers* *Curators* *Commentators* *Viewers*	[Tinati et al., 2012] *in Twitter*
Diffuser *Gatekeeper* *Information broker*	[Tyshchuk et al., 2013] *in Twitter*	*Angels* *Active Learners* *Passive Learners* *Social Networkers* *Journalist-Narcissists* *Screamers*	[Maulana and Tjen, 2013] *in Facebook & Twitter*
Outlier role *Principal role:* *Community leader* *Global principal* *Peripheral role* *Mediator role* *Extrovert role*	[Fagnan et al., 2014]	*Celebrity* *Newbie* *Lurker* Unhealthy roles: *Flamer* *Ranter* *Troll*	[Golder and Donath, 2004] *in Usenet*
References *Engaged leaders* *Mediators* *Diversified listeners* *Listeners*	[Beguerisse-Díaz et al., 2014] *in Twitter*	*Leaders* *Motivators* Non-supportive: *Chatters*	[Nolker and Zhou, 2005] *in Usenet*

2.3 Influence and trust techniques

As noted in the previous section, influence and trust are prominent SNA issues for socially-oriented efforts such as SmartH2O, because detecting influential and trustable users within a real or virtual community may help targetting them with ad hoc messages and thus optimize the dissemination of the user's data collection tasks and sustainability actions envisioned by the project.

This section deepens the survey of role and interpersonal relationship analysis in social media, by focusing specifically on the definition of trust and influence and on the operationalization of these concepts introduced by social media analysis by means of a series of metrics, computable from social network data.

2.3.1 Trust techniques

The concept of trust has been studied in sociology [Molm et al., 2000], psychology [Cook et al., 2005], economics [Huang, 2007] and computer science [Maheswaran et al., 2007]. Moreover, during the last years many companies have emerged in the market having as core business to analyze social media and extract various statistics about users and content, e.g., Blogmeter, Klout, PeerIndex and ProSkore. These statistics include online reputation,

influence evaluation, engagement and content relevance. Usually, the offered services are grouped in software suites that allow one to analyse dynamically content (e.g., by filtering it by topic and relative subtopics, or by extracting the most diffused terms and concepts for a specific topic) and users (e.g., by visualizing who are the most active users in the field).

The most common strategies for influence evaluation in social media are: i) either to identify influencers, or ii) to study the maximization of influence spread in a social network. In [Kiss and Bichler, 2008] the identification of influencers is achieved by considering the structural properties of networks. In [Lu et al., 2012] a graph-based framework is used to predict the evolution of influencers. [Scripps et al., 2009] investigated how different decisions such as selection and influence affect the dynamics of social networks. [Gomez Rodriguez et al., 2010] developed a method to trace paths of diffusion and influence through networks. Furthermore, some researchers investigated the problem of maximizing influence on a person network (ego-net) for applications such as viral marketing [Domingos and Richardson, 2001,Kempe et al., 2003,Goyal et al., 2010]. In [Tan et al., 2010], authors studied how to track and predict users' action according to a learning model. However, these works neither consider heterogeneous information nor learn topics and influence strength jointly: there are no works in the state of the art that analyze the full spectrum of multimedia content produced and consumed by users to estimate a local and contextual notion of trust. Moreover, they did not consider the topic-level influence: in most of these works, a user is influencer if she is interesting for a large part of users, independently from the topic of interest one is tracking. However, this falls on the million follower fallacy, where a user is influential if she is a celebrity.

In the state of the art, patterns of temporal variation of popularity have been investigated too, mostly focusing on the attention received by pieces of content. Previous works include for instance the study of video popularity saturation on YouTube in relation to content visibility [Figueiredo et al., 2011] and the classification of bursty Twitter hashtags in relation to the volume of related tweets before and after the peak [Lehmann et al., 2012]. Time series have been used to predict popularity in blogs, where the reaction time of the crowd is strongly correlated to the expected overall popularity [Lerman and Hogg, 2010]. However, a few works focus on the mining of temporal patterns in content diffusion and people activity on social media, which could help in tracing the dynamics of influence.

Another aspect regards multimedia search. Before the social media era, multimedia search aimed at answering multimedia queries (using e.g., query by example approach) on static databases. Social media change the scene by generating and sharing huge volumes of ephemeral content. The volumes of image/video shared through the social media every day and the ephemeral nature of the postings require new indexing and searching methodologies. It is thus necessary to design and develop a NoSQL-based time-aware multimedia search service that is able to answer complex time related and multimedia queries, supporting large but ephemeral multimedia content processing.

2.3.2 Influence and people search techniques

Modern influence analysis methods have concentrated efforts on the analysis of Twitter data, because such social media platform is more open to data collection than other ones.

Influencers in Twitter are mainly celebrities, popular bloggers and organizations [Zhai et al., 2014,Cataldi and Aufaure, 2014,Bi et al., 2014] that lead discussions mostly on topics such as fashion, music, etc. that could be labelled as entertainment [Cataldi and Aufaure, 2014]. Moreover, Twitter users tend to strengthen the relationships with users in the same areas of interest. Thus, their retweet connections with similar users allow to identify and discriminate their main area of influence [Cataldi and Aufaure, 2014].

It is evident that each estimated influence value is strictly dependent from the considered topic-based community: National Geographic has a higher influence value on the scientific do- main than the one of Barack Obama on political news when considering number of retweets, since a larger number of authoritative information sources retweeted National Geographic [Cataldi and Aufaure, 2014]. Highly connected users and/or community can easily result in higher estimated influence values in their domains of interest [Cataldi and

Aufaure, 2014].

On the other hand, passive users (i.e., people who follow many people but retweet a small percentage of the information they consume) are robot accounts (which automatically aggregate keywords or specific content from any user on the network), suspended accounts (which are likely to be spammers) and users who post extremely often. Moreover, the amount of attention a person gets may not be a good indicator of the influence they have in spreading the message, and users with very low number of followers often have high influence [Romero et al., 2011].

There is evidence that influence changes over time, so that the group of top-10 influencers change frequently, leaving space for other people to become influencers for the same topics [Cha et al., 2010].

2.3.2.1 Influence metrics

Several metrics were proposed in the literature for identifying influencers in a social network. Such metrics try to quantify the degree of influence of a given user, and are mainly based on: i) some descriptor of the content of the user posts; ii) some descriptor of the social network neighbourhood of the user. Generally, the multimedia content (e.g., photos contained in the posts) is not considered.

In the following, we consider the use case of Twitter. Similar metrics may be found for the other social media, but since Twitter is the most analysed source in this field of research, we can focus on it.

2.3.2.1.1 Text-based metrics

In this Subsection, we illustrate the most widely used descriptors of the text of the user posts.

2.3.2.1.1.1 Analysis of original tweets of a user

The most used descriptors of the posts text are be the number of original tweets, the number of shared URLs, the number of used keywords and hashtags [Pal and Counts, 2011,Jabeur et al., 2012], which come natural. Moreover, when the analysis focuses on a specific topic, one could decide to measure the self-similarity score of a user, which measures how similar are the author's recent tweets with respect to her previous tweets: if the author focuses himself on a topic, then the self-similarity of her posts is expected to be high [Pal and Counts, 2011].

Furthermore, some works [Weng et al., 2010] take into account the homophily of different authors: users that talk about similar topics are easily involved in what the other is saying, while low homophily profiles do not share topics of interest.

Finally, since users could produce text containing typos, some works [Cataldi and Aufaure, 2014] compute all the possible n-grams (i.e., all the possible combinations of characters of every word in the original tweet text) and use them as a descriptor of the produced text. This introduces more robustness on the typos in the text, since at least one of the produced n-grams will contain the correct, typo-free term.

However, there are some works which deviate from the typical approach. An example can be found in a quite recent work [Quercia et al., 2011] that evaluates the influence of an author by estimating her behavioral traits. This work uses the LIWC dictionary to extract language categories which are typical of some personality traits (e.g., self-esteem, self-confidence). Then, for each tweet one can extract the percentage of words that describe each language category: if the most shown traits are typical of an influential person, then the user is considered an influencer.

2.3.2.1.1.2 User involvement

A mention in the form of @user captures a user attention to follow the content published by the author. Some works [Pal and Counts, 2011,Jabeur et al., 2012,Cha et al., 2010,Lian et al., 2012] use this factor as a metrics for stating how much a user is able to involve others in the topic. The degree of involvement is measured by taking into account, for instance, the

number of mentions of others by the authors and the number of users mentioned by the authors.

2.3.2.1.1.3 Conversational degree of a user

Some works in the state of the art measure the conversational degree of a user on Twitter, based on the presence of mentions in the tweet text [Pal and Counts, 2011,Lian et al., 2012].

A conversational tweet is a tweet directed to other user. To create such tweets the authors put the mention @user before the tweet text, meaning that the tweet is directed to the user @user. The conversational degree of a user is thus generally computed depending on the number of conversational tweets produced by the user, the number of conversations started by the user (i.e., those conversations whose first conversational tweet was produced by the user) and the number of conversational tweets that involve the user.

2.3.2.1.1.4 Content replication

A retweet on Twitter is the copy of forwarding of a user's posts by other users. To create such tweets the authors put the string RT @user, meaning that the tweet is copied from the user @user (i.e., the original author of the post is @user).

Several works consider the content replication degree (i.e., the capability of diffusing the content on Twitter) as a possible descriptor of the influence of a user [Pal and Counts, 2011, Jabeur et al., 2012,Lian et al., 2012,Cataldi and Aufaure, 2014,Cha et al., 2010,Kong and Feng, 2011]. This degree is measured in terms of the number of tweets the user copies from others, the number of tweets others copy from the user, the number of users that were involved in retweeting operations with the user and the number of topic-related retweets.

2.3.2.1.1.5 Hybrid metrics

Some works [Pal and Counts, 2011,Zhai et al., 2014] propose ways of aggregating the factors explained so far, which, by combining simple statistics on the tweets (such as number of retweets, number of conversational tweets, etc.), measure the topical focus of an author, her retweet impact, her ability of diffusing content etc.

2.3.2.1.2 Neighbourhood-based metrics

In this Subsection, we introduce the most widely used descriptors of the social network neighbourhood of a user's social graph.

The influence of a user depends on the structure of her social network: the larger it is, the higher is the probability that the information is diffused and shared. Several works [Zhai et al., 2014,Pal and Counts, 2011] capture the extension of a user's social network by considering the number of her followers (i.e., the users reading her contents) and friends (i.e., the users whose content is read by her). However, not all the followers and friends are interested in the content the user publishes, since their focus could be other topics. Thus, a more refined analysis [Cha et al., 2010,Agarwal et al., 2008] filters the number of followers and friends so as to consider just the ones that talk about the topic on which the user in analysis is focus.

When talking about influence on the social network of a user, one could consider two important factors: homophily and reciprocity [Kwak et al., 2010]. Homophily is the similarity between a user and her followers and friends: it states how much the topics, the geographic position and the popularity degree of the friends/followers in are similar to the ones of the user in analysis. Reciprocity, on the other hand, is the property for which a user in the social network follows another user just because that user followed her. While homophily is a good descriptor of how much two users in a social network are close (i.e., are similar), reciprocity is not a good descriptor of the relevance of a user for another, since it is just an expression of politeness towards one's followers.

Other works use other mathematical properties of graph structures to measure the level of influence of users. For instance, the works [Shetty and Adibi, 2005,Sun and Ng, 2013] measure the influence of a user u by removing her from the graph G(U): if the difference between the graph entropy of G(U) and the graph entropy of G({U \ u}) is high, then this

means that u has high influence on the graph structure. In other cases, , the influence of a user on another user is measured by computing their distance on the network, which can be expressed either as a sum of the weights connecting the two nodes in the network (where the weight depends on the relationship type, i.e., friend, follower, not in relation) [Weitzel et al., 2012], or simply by the number of edges separating the two nodes [Cataldi and Aufaure, 2014].

Finally, several works [Pal and Counts, 2011,Zhai et al., 2014,Kazienko and Musial, 2007,Sun and Ng, 2013,Agarwal et al., 2008] build aggregated metrics that compute the information diffusion, the topical follower signal, the social position, the relationship strength and the topological influence of a user in her network.

2.3.2.1.3 Other metrics

Other works build descriptors based on other factors. For instance, the work in [Chen et al., 2014] checks if the profile of a user is verified, meaning that she corresponds with high probability to a celebrity and thus has a high influence. Moreover, the same work applies sentiment analysis techniques to state whether the user is talking good or bad about a specific topic. Indeed, usually fans talk well about a subject, while experts criticize it. Detecting a high expertise level of a user is a suggestion of the fact that he is an influencer on the topic.

2.3.2.2 Influencers detection algorithms

The abovementioned metrics are embedded within data processing methods for extracting the influencers from social network data sets. In the following, we overview the main approaches and the algorithms used as baselines in the literature.

2.3.2.2.1 Score computation

The simplest influencers retrieval approach used in the state of the art is the one of computing an influence score for each user, using one of the abovementioned metrics, and then returning the top-K users with the largest influence score [Agarwal et al., 2008,Kong and Feng, 2011,Bi et al., 2014].

2.3.2.2.2 Clustering and classification

Some works perform user clustering based on user influence characteristics, to find groups of users having similar influence on the network [Pal and Counts, 2011, Chen et al., 2014]. In other cases, users are classified in influence classes (e.g., influential, popular, listener, highly read) according to their features (both text-based and graph- based) [Quercia et al., 2011].

2.3.2.2.3 Graph structure analysis

In the state of the art, some works that classify the importance of a node with respect to the graph topology can be found too [Shetty and Adibi, 2005]. Here, each node n_i is temporarily removed from the graph G, and the graph entropy gap between G and G \ n_i is computed. Then, the node n_i with the largest entropy gap is selected as the most influential, since it causes the largest impact on the graph. Similar works analyze the network to find those users with the largest ability of spreading news and content over the network [Saez- Trumper et al., 2012]. Other works create algorithms inspired to PageRank [Jabeur et al., 2012,Cataldi and Aufaure, 2014,Weng et al., 2010].

2.3.2.2.4 Examples of algorithms

In the following, we list some algorithms used by other authors as baselines for influencers retrieval.

The work by [**Pal and Counts, 2011**] uses the same retrieval algorithm, although the influence metrics is downgraded to a simpler version (e.g., when the metrics considers both graph-based and text-based metrics, baselines could be defined as the same metrics, in which either the graph-based or the text-based dependence is relaxed).

Network centrality metrics [Huang et al., 2013] (e.g., degree, closeness) and graph-based characteristics [Jabeur et al., 2012,Kwak et al., 2010] (e.g., number of followers [Weng et al., 2010], number of retweets) can be used as baselines to estimate the user importance. The approach in [**Cha et al., 2010**] presents an empirical analysis of influence patterns in a popular social medium. Using a large amount of data gathered from Twitter, three different measures of influence are compared: indegree, retweets, and mentions. Indegree influence represents the number of followers of a user and directly indicates the size of the audience for that user. Retweet influence, measured through the number of retweets containing one's name, indicates the ability of that user to generate content with pass-along value. Mention influence, which through the number of mentions containing one's name indicates the ability of that user to engage others in a conversation. The approach examines how these three types of influence perform in spreading popular news topics, focusing on different topics. It also investigates the dynamics of an individual's influence by topic and over time. The approach proposes a characterization of the precise behaviors that make ordinary individuals gain high influence over a short period of time.

Several works use the **PageRank** algorithm [Page et al., 1999] as baseline [Romero et al., 2011,Saez-Trumper et al., 2012,Kong and Feng, 2011,Huang et al., 2013,Kwak et al., 2010,Weng et al., 2010] or a variant of the same algorithm [Jabeur et al., 2012]. Other works [Kong and Feng, 2011] consider the HITS algorithm as baseline.

TwitterRank [Weng et al., 2010] is an extension of PageRank algorithm proposed to measure the influence of users in Twitter. It is an approach that measures users' influence taking both the topical similarity between users and the link structure into account. In TwitterRank, a random user visits another user with a certain topic-specific probability by following the appropriate edge in the social network graph. A set of topic-specific vectors is generated, which measures the influence of the users over individual topics. An aggregation is used to obtain the overall influence of a user. The work also reports that there exists homophily in Twitter, which implies that a twitterer follows a friend because she is interested in some topics the friend is publishing, and the friend follows back because she finds they share similar topical interest.

The work in [**Saez-Trumper et al., 2012**] presents a ranking strategy that focuses on the ability of some users to push new ideas that will be successful in the future. These users are denoted as trendsetters. To achieve that, it combines temporal attributes of nodes and edges of the network with a Pagerank based algorithm to find the trendsetters for a given topic. In order to be able to identify persons that spark the process of disseminating ideas that become popular in the network, timing information is introduced on the social graph. The dissemination of information is modelled as a topic-sensitive weighted innovation graph, where a topic represents a collection of trends. This graph provides key information to understand who adopted a certain topic that triggered attention of others in the network. The algorithm shows that nodes with high in-degree tend to arrive late for new trends, while users in the top of the ranking tend to be early adopters that also influence their social contacts to adopt the new trend.

Finding trustable users in a social network can be related to a topic or not; it depends on the content they produce but also on the number of people that reaches this content which depends on how well a user is connected. Large quantities of content are produced daily over social networks. Therefore, it is challenging to find the most notable authors of content.

Works in which the topic of the tweets is estimated from the data use other (simpler) classification algorithms (e.g., Naive Bayes classifiers, K-nearest neighbors classifiers) as baselines [Cataldi and Aufaure, 2014].

An algorithm that determines the influence and passivity of users based on their information forwarding activity is investigated in [**Romero et al., 2011**]. The proposed work performs an analysis of the propagation of web links on Twitter over time to understand how attention to given users and their influence is determined. The approach builds a general model for influence using the concept of passivity in a social network, i.e. influential users must overcome user passivity. The devised influence measure utilizes both the structural properties of the network as well as the diffusion behavior among users. The influence of a

user thus depends not only on the size of the influenced audience, but also on their passivity. The passivity of a user is a measure of how difficult it is for other users to influence her. The user's influence score depends on the number of people she influences and their passivity, as well as how dedicated the people she influences are. The user's passivity score depends on the influence by those who she is exposed to, but not influenced by, as well as, how much she rejects other user's influence compared to everyone else.

2.3.3 Results evaluation

Influence is a subjective measure, and thus multiple ways of assessing the performance of the proposed metrics and algorithms are proposed in the state of the art.

2.3.3.1 Basic metrics

Some works use some trivial metrics to state the accuracy of their work. For instance, the work in [Weng et al., 2010] computes the accuracy measure by intersecting the obtained result set (i.e., set of influencers) with the set of most active users in the dataset. The implicit assumption here is that an active user is also an influencer, which in some cases can be true. However, since this does not always hold, this metrics is too simple to capture an intricate measure such as the influence (which depends on several factors).

2.3.3.2 Manual evaluation

Several works propose evaluation metrics based on manual evaluation of the result set [Pal and Counts, 2011,Shetty and Adibi, 2005,Cha et al., 2010,Huang et al., 2013, Zhai et al., 2014, Bi et al., 2014, Cataldi and Aufaure, 2014, Weng et al., 2010]. This evaluation procedure is based on the manual check of the influencer profile, to see whether it can be considered as an influencer by humans. For instance, if the topic is the movie 'Toy Story 3 ', then the director of the same movie can be considered as an influencer for the topic.

2.3.3.3 User study

Other works [Pal and Counts, 2011,Hannon et al., 2010,Jabeur et al., 2012,Chen et al., 2014] perform users studies with numerical evaluation of the topic relevance and influence metrics. In these studies an annotator (which usually is a specific expert in the field) or a set of annotators are required to go through the set of results, to assess its quality. Evaluations can be either anonymous (e.g., only the text is shown to the annotator, and thus the influencer profile is not visible) or non-anonymous (e.g., the name of the user is known and thus the profile is visible). In case of multiple annotators, the opinions are aggregated in a unique evaluation, using classical methodologies (e.g., majority voting).

2.3.3.4 Manual ground truth

When the focus is the one of classifying tweets in topics [Cataldi and Aufaure, 2014], manual ground truth construction is performed on the collected dataset, so as to compute the accuracy at the end of the classification process. This can be generally done when a fixed dataset or a fixed training set is crawled from the social network.

2.3.3.5 Top-K posts on other services

Some work use other services to evaluate their topic classification quality. These services are usually news aggregators whose aim is to select viral Internet issues. Consequently, the top-K posts one can find on those services correspond to the K posts with the highest visibility on the network. For instance, the work in [Cataldi and Aufaure, 2014] compares its results with the top-K posts on Digg. Another example can be found in [Kwak et al., 2010], which compares its result set with the one of Google Trends (i.e., a collector of trending topics by Google) and CNN Headlines.

Other works use the same principle, but exploiting information that is already present in the analyzed social network. For instance, [Bi et al., 2014] computes the accuracy of its results by counting the number of verified profiles that are present in the retrieved result set, since the verified users are considered celebrities (and thus relevant users).

2.3.4 Influence maximization

A social network, the graph of relationships and interactions within a group of individuals, plays a fundamental role as a medium for the spread of information, ideas, and influence among its users [Kempe et al., 2003]. Influence detection in social networks includes approaches for community detection, influential users detection, trust among users, and influence propagation/maximization within a social network. As a result of the users' interactions, cohesive groups (communities) are formed within a social network. Users tend to interact more within a community than between communities.

Influence maximization, defined by Kempe, Kleinberg, and Tardos in [Kempe et al., 2003], is the problem of finding a small set of seed nodes in a social network that maximizes the spread of influence under certain influence models. Influence maximization is particularly interesting to many companies as well as individuals that want to promote their products, services, and innovative ideas through the powerful word-of-mouth effect [Chen et al., 2009]. In these models, each individual node of the social network graph is being either active (an adopter of the innovation) or inactive [Kempe et al., 2003]. Models for the processes of influence maximization have been extensively studied in a number of domains. These models have two assumptions: i) the tendency of each node to become active increases monotonically as more of its neighbors become active; ii) nodes can switch from being inactive to being active, but do not switch in the other direction.

The two basic and most studied diffusion models in the literature are the **Linear Threshold Model** and the **Independent Cascade Model**.

The **Linear Threshold Model** has been investigated in its multiple flavours [Berger, 2001; Granovetter, 1978; Young, 2006; Peleg, 1997], but its core is explained in the following paragraph. In the Linear Threshold Model, as defined in [Kempe et al., 2003], a node is influenced by each neighbour according to a specific weight. Each node chooses a threshold value uniformly at random. This represents the weighted fraction of the node's neighbours that must be active so that the node itself can become active. Given a random choice of thresholds and an initial set of active nodes, the diffusion process unfolds itself deterministically in discrete steps. In time instant t, all nodes active in the previous time instant t-1 remain active, plus the nodes that become active as a result of the total weight of their active neighbors which is at least the assigned threshold value. The random selection of the threshold values models the lack of knowledge of their real values.

In the **Independent Cascade Model**, introduced in [Goldenberg et al., 2001a, 2001b] in the context of marketing, the process of node activation is unfolded in the following discrete steps: In each time instant t, when a node becomes active, it has a single chance to activate each of its inactive neighbors with a given probability value (a system parameter). If the node succeeds in the activation, then the newly activated nodes become active in the next time instant t+1 [Kempe et al., 2003]. If the node did not succeed in activating the neighbors, it cannot make further attempts to activate them in the subsequent steps. The process runs until no more activations are possible.

A broader framework that simultaneously generalizes these two models is given in [**Kempe et al., 2003**] which allows one to explore the limits of models in which strong approximation guarantees can be obtained. In the proposed general threshold model, a decision whether the node will become active depends on an arbitrary monotone function of the set of its neighbors that are already active. The diffusion process follows the general structure of the Linear Threshold Model, such that, the node becomes active if the monotone function of its active neighbors is greater than the threshold value.

The **General Cascade Model** allows the probability that a node succeeds in activating one of its neighbors to depend on the set of the neighbors that have already tried and failed to activate it. The general cascade process works in the same way as the Independent Cascade Model. The general framework has equivalent formulations in terms of thresholds and cascades, thereby unifying these two views of diffusion through a social network. These two models are equivalent, and they can be converted between themselves.

SIMPATH [Goyal et al., 2010] is an algorithm for influence maximization under the Linear

Threshold Model. SIMPATH incorporates three key novel ways of optimizing the computation and improving the quality of seed selection, where seed set quality is based on its spread of influence: the larger its spread, the higher its quality. The algorithm computes the spread by exploring simple paths in the neighbourhood. SIMPATH leverages two optimizations. The VERTEX COVER OPTIMIZATION cuts down the spread estimation calls in the first iteration, while the LOOK AHEAD OPTIMIZATION improves the efficiency in subsequent iterations.

The approach in [**Chen et al., 2010**] proposes an **Independent Cascade Model** based algorithm for influence maximization in large-scale social networks. The heuristic gains efficiency by restricting computations on the local influence regions of nodes. Moreover, by tuning the size of local influence regions, the heuristic is able to achieve tunable tradeoff between efficiency (in terms of running time) and effectiveness (in term of influence spread). The algorithm first computes maximum influence paths (MIP) between every pair of nodes in the network via a Dijkstra shortest-path algorithm, and ignore MIPs with probability smaller than an influence threshold, effectively restricting influence to a local region. Then it performs union of the MIPs starting or ending at each node into the arborescence structures (a tree in a directed graph where all edges are either pointing toward the root (in-arborescence) or pointing away from the root (out-arborescence)), which represent the local influence regions of each node. The approach considers only the influence propagated through these local arborescences, and this model is called the maximum influence arborescence (MIA) model.

The method in [**Chen et al., 2011**] investigates a new influence cascade model, the Independent Cascade Model with negative opinions (IC-N) which explicitly incorporates the emergence and propagation of negative opinions into the influence cascade process. The IC-N model is associated with a new parameter called the quality factor. Informally, the IC-N model works as follows. Initially, a set of nodes in the network is selected as seeds and are activated (e.g. provided with free trials of the product/service). Each seed turns positive (experiencing good quality of the product/service) with a probability q and with probability 1 – q turns negative (encountered defects). At each time step, a positively activated node in the previous step tries to positively activate each of its non-active neighbours, and if successful (with a success probability) the neighbour is activated (bought the product/service). Meanwhile a negatively activated node in the previous step also tries to negatively activate its non-active neighbors, and if successful the neighbors become negative (accepted negative opinions and avoiding the product/service). If several nodes try to activate the same node in one step, the order of activation trials is random. In order to maximize the influence of the active nodes, the approach focuses on maximizing the expected number of positive nodes in the network after the cascade (positive influence spread).

CELF [Leskovec et al., 2007] is an efficient algorithm for influence maximization that scales to large problems, achieving near optimal placements. It is based on the concept of submodularity, such that the idea is to select a small set of nodes which will maximize the propagation of influence (information cascades) throughout the network. These information cascades initiate from a single node of the network, and spread over the graph, such that the traversal of every edge takes a certain amount of time (indicated by the edge labels). Every placement of nodes is associated with a cost which should not exceed a specified budget that can be spent. CELF achieves near-optimal placements of nodes (guaranteeing at least a constant fraction of the optimal solution), providing a novel theoretical result for non-constant node cost functions. CELF develops online bounds on the quality of the solution obtained by any algorithm.

The approach in [**Chen et al., 2009**] presents a greedy algorithm that tackles the efficiency problem of influence maximization. It is based on a specifically designed scheme combined with the CELF optimization. From the CELF algorithm, it uses the submodularity property of the influence maximization objective to greatly reduce the number of evaluations on the influence spread of vertices. The approach also introduces new degree discount heuristics with influence spreads that are significantly better than the classic degree and centrality-based heuristics and are close to the influence spread of the greedy algorithm. The proposed greedy algorithm and degree discount heuristics are derived from the Independent Cascade Model and weighted cascade model.

The technique proposed in [**Borgs et al., 2014**] investigates a constant-factor approximation algorithm for the influence maximization problem, under the standard Independent Cascade Model of influence spread, that runs in quasilinear time. The runtime of the algorithm is independent of the number of seeds, which is essential when the relevant input networks are massive. The algorithm applies a random sampling technique to generate a sparse hypergraph representation of the network. Each hypergraph edge corresponds to a set of individuals that was influenced by a randomly selected node in the transpose graph (the original network with all the edge directions reversed). The hypergraph encodes the influence estimates: for a set of seed nodes, the total degree of the seed set in the hypergraph is proportional to the influence of the seed set in the original graph. In the second step, a standard greedy algorithm is run on this hypergraph to return a node set of approximately maximal total degree. The algorithm can also be modified to run in sublinear time, with a correspondingly reduced approximation factor.

Linear Influence Model for influence diffusion is introduced in [Yang and Leskovec, 2010]. In this model, rather than requiring the knowledge of the social network and then modelling the diffusion by predicting which node will influence which other nodes in the network, the focus is on modelling the global influence of a node on the rate of diffusion through the (implicit) network. The number of newly activated nodes is modelled as a function of which other nodes were activated in the past. For each node, an influence function is estimated, that quantifies how many subsequent activations can be attributed to the influence of that node over time. A nonparametric formulation of the model leads to a simple least squares problem that can be solved on large datasets.

The approach in [**Barbieri et al., 2013**] studies social influence from a topic modelling perspective. As a result, it introduces topic-aware influence-driven propagation models that are more accurate in describing real-world cascades than the standard (i.e., topic-blind) propagation models such as the Independent Cascade and Linear Threshold models. The approach first proposes simple topic-aware extensions of these models.

In the **Topic-aware Independent Cascade** (TIC) model, the node probabilities depend on the topic, such that the probability represents the strength of the influence of a node on its neighbor on a specific topic. For each item that propagates in the network, its distribution over the topics is given. The propagation happens as in the Independent Cascade Model.

In the **Topic-aware Linear Threshold** (TLT) model, a weight is assigned to each arc and for each topic, such that the sum of incoming weights for each node and each topic does not exceed 1. The propagation for a topic happens in the same way as for the Linear Threshold Model.

Due to the limits of the TIC and TLT models, a new influence propagation model is introduced called AIR (**Authoritativeness–Interest–Relevance**). Instead of considering user-to-user influence, the proposed model focuses on user authoritativeness and interests in a topic, leading to a drastic reduction in the number of parameters of the model, with benefits in terms of reduced risk of overfitting and reduced learning time. A generalized expectation maximization (GEM) approach is devised to learn the parameters that maximize the likelihood for the AIR model.

Spine (Sparsification of Influence Networks) [Mathioudakis et al., 2011] is an efficient algorithm for finding the "backbone" of an influence network. It is based on sparsification, a data reduction operation that allows better data visualization, digestion and interpretation. Given a social graph and a log of past propagations, we build an instance of the independent-cascade model that describes the propagations by eliminating a large number of links, and preserving only those that are important in the information propagation (set of links that maximize the likelihood of observed data). The aim is to reduce the complexity of the Independent Cascade Model, while preserving most of its accuracy in describing the data. SPINE has two phases: in the first phase it selects a set of arcs that yields a finite log-likelihood; in the second phase, it greedily seeks a solution of the maximum log-likelihood. The found solution is guaranteed to be close to the optimal one.

The work in [**Goyal et al., 2010**] proposes a method for building influence models based on a social graph and a log of users' actions. The approach is based on a propagation graph

whose nodes are users that performed an action with edges connecting them in the direction of the propagation. When a user performs an action, a node is activated with respect to that action. Once it is activated, it cannot be deactivated anymore. The power to influence the nodes is modelled as influence probability. The problem tackled in this approach is how to learn influence probabilities among the users, by mining the available set of past propagations. The adopted framework is based on the General Threshold model. The approach proposes 3 types of models that capture individual influence, expressed as a probability of a user and its neighbor(s), used to compute the joint influence. The first class of models assumes the influence probabilities are static and do not change with time. The second class of models assumes they are continuous functions of time. The evaluation showed that continuous time models are by far the most accurate, but they are very expensive to test on large data sets. Thus, the approach proposes an approximation known as Discrete Time Models where the joint influence probabilities can be computed incrementally and thus efficiently. The approach also develops techniques for predicting the time by which a user may be expected to perform an action.

2.3.5 Trust computation

The computational problem of trust is to determine how much one person in a social network should trust another person to whom they are not connected [Golbeck, 2005]. Trust involves a belief that the trusted person will take an action that will produce a good outcome. The idea of trust inference is to recommend to a node of a social network how much to trust another node that it is not connected to. As with all social relationships, it is difficult to quantify trust since its properties are fuzzy.

TidalTrust [Golbeck, 2005] calculates trust recommendations in networks with continuous values. TidalTrust collects trust data from all referral paths with the shortest length from a source to a sink. Calculations move forward from a source to a sink in the network, and then pull back from the sink to return the final value to the source.

The algorithm selects referral paths with strength above a threshold and uses them to compute the overall trust value.

SUNNY [Kuter and Golbeck, 2007] is an algorithm for trust inference based on probabilistic confidence models. The approach takes a trust network, represented as a graph, and produces a Bayesian Network suited for approximate probabilistic reasoning. SUNNY performs a probabilistic logic sampling procedure over the Bayesian Network. To do so, it computes estimates of the lower and upper bounds on the confidence values, which are then used as heuristics to generate the most accurate estimates of trust values of the nodes of the Bayesian Network.

The approach in [**Golbeck, 2009**] explores the relationship between trust and profile similarity. Surveys and analysis of data in existing systems show that when users express trust, they are capturing many facets of similarity with other users.

A study presented in this work, where users are given generated profiles for hypothetical users, demonstrates that several features of profile similarity correlate with trust. These results are then brought to data to show that using that set of features to predict trust is better correlated with known trust values, and is more accurate than using overall similarity alone.

The approach also discovers a correlation between trust and the largest single difference in ratings, and between trust and the agreement on items the source has given extreme ratings. Some sources tend to assign higher ratings than others when rating a population of sinks that vary from the source in the same way.

CertProp [Hang et al., 2009] is an evidence-based approach, that provides efficient operators, concatenation (deals with propagation of trust ratings across a path), aggregation (deals with combination of trust ratings from paths between the same source and target), and selection (chooses the most trustworthy path to each node) that can propagate trust accurately. These operators satisfy useful algebraic properties. The approach motivates a new way to transform subjective opinions into objective evidence. It experiments with two types of transformations: linear based on normalization, where a belief from a higher rating should also be higher; and Weber-Fechner, where the relationship between stimulus (good

experience) and perception (opinion ratings) is logarithmic. These transformations also follow the idea that the average opinion yields the lower certainty of transformed trust. It helps reduce the subjectivity in opinion-based datasets so that the evidence-based approaches like CertProp can apply.

2.4 Evaluation of the social network analysis techniques

In this section we evaluate the approaches most relevant to SmartH2O among the many proposals surveyed in Sections 2.1, 2.2 and 2.3.

2.4.1 Influence metrics

Influence metrics play a central role in influencer detection, as they operationalize the notion of influence into a measurable property.

Table 4 examines the most relevant metrics surveyed in Section 2.3 under the perspective of four evaluation dimensions:

- **Language independence**: whether or not the computation of the metrics requires natural language processing; this dimension affects the portability of the metrics computation software to international domains, where the basic libraries for NLP may be less accurate than for mainstream languages (most notably English) or unavailable.
- **Topic-relevance discovery**: whether or not the computation of the metrics has the collateral benefit to help understand the subject matter on which influence is detected, e.g., generic environmental activism with respect to water-specific activism.
- **Communication skills evaluation:** whether or not the computation of the metrics can help understanding the communication.
- **Content diffusion evaluation:** whether or not the computation of the metrics supports the computation of the influencer's reach of communications, e.g., in space (geographical reach, local or global) and time (e.g., one shot, burst communication, or enduring conversations and debates).
- **Connection with other users:** whether or not the computation of the metrics has the collateral benefit of extracting the subnetworks that constitute the neighborhood of the influencer, which can be used to perform sub-community detection.

Table 4. Evaluation of influencer detection metrics.

Metrics	Language independence	Topic-relevance discovery	Communication skills evaluation	Content diffusion evaluation	Connection with other users
Text based metrics		X	X	X (retweet)	
Graph based metrics	X			X (edges)	X
Profile reputation (e.g., via verified accounts)	X				X
Expertise level inference via sentiment analysis		X	X	X	

2.4.2 Algorithms for influencers retrieval

Beside the metrics applied for the evaluation of influence, also algorithms vary and exhibit different characteristics, which make them more or less suitable for a given SNA goal.

Table 5 shows how different algorithms position with respect to the evaluation perspectives of:

- **Diffusion of information**: whether or not the algorithm is suitable to a context where maximising the diffusion of information through influencers is the principal goal.
- **Similarity of users**: whether or not the algorithm supports the discovery of influential users with similar characteristics, e.g., topical relevance.
- **Scalability on large graphs**: whether or not the algorithm is intrinsically conceived for working on the very large graphs that are typical of open-ended social network analysis on mass scale social platforms or conversely specialize for focused searches on smaller scale communities.
- **Influence propagation tracking**: whether or not the algorithm support tracking in space and time the effects of communications though influential users.

Table 5. Evaluation of influencer detection algorithms

Algorithms	Diffusion of information	Similarity of users	Large graphs	Influence propagation
Score computation	X (metrics that mix graph-based features and content-based features)	X (based on published content)	X	
Clustering and classification		X		
Graph structure analysis • PageRank based • Centrality metrics	X			X

2.4.3 Evaluation of Community Detection Methods and Community Role Schemes

In this section we provide an evaluation of the algorithms and the approaches with respect to defined criteria that are considered important in the domain of application.

Regarding graph community detection algorithms, some of the most commonly employed were described earlier in 2.2.1.1. Several studies have already provided useful knowledge with respect to their evaluation (e.g. [Xie et al., 2013]), based on which we provide here a short overview.

One of CPM's limitation is that it assumes that the graph has a large number of cliques, and thus it can fail to give meaningful covers for graphs with just a few cliques, like some social networks whereas in the case of many cliques the method may deliver trivial community structure, like a cover consisting of the whole graph as a single cluster [Fortunato, 2010]. CPM has appeared suitable for networks with dense connected parts and to give good results for small values of *k* [Xie et al., 2013]. The available implementation (CFinder) has polynomial time complexity in many applications [Palla et al., 2005] but it fails to terminate in many large social networks [Xie et al., 2013].

OSLOM has been very popular to the relevant literature studies and presents advantages that often contribute to this preference. The reasons, for example, why it was selected as the main method in [Grabowicz et al., 2012] were the capacity to analyse a full directed follower network in a reasonable time, the detection of overlapping communities as well as nodes belonging to none of the groups (or singleton communities), the fact that the clusters are

statistically significant according to a null model and fact that its implementation is publicly available. OSLOM usually generates a significant number of singleton communities [Xie et al., 2013] which maybe a more realistic structure in the typical case of a networks created with Breadth-first search where it is very likely to find nodes weakly connected. However, it can produce unstable results as demonstrated also in [Lancichinetti and Fortunato, 2012] and thus they proposed consensus clustering.

SLPA has appeared to perform well and can be also adapted for weighted and directed networks by generalizing the interaction rules, known as SLPAw [Xie et al., 2013]. However, one of the weaknesses of the method is that it doesn't assume singleton communities.

[Xie et al., 2013] performed an extensive review of the state of the art in overlapping community detection algorithms comparing 14 different algorithms. Overall both SLPA and OSLOM outperformed pother methods in cases of low overlapping density networks, while for networks with high overlapping density and high overlapping diversity SLPA provides relatively stable performance. However, it is important to note that the reviewed work was mostly on unweighted networks.

In terms of application of the methods in the specific domain of approaches in Twitter, OSLOM has been applied in [Grabowicz et al., 2012; Greene et al., 2012], CPM was used in [Java et al., 2007; Lim and Datta, 2012a, 2012b, 2012c, 2013] and SLPA in [Deitrick and Hu, 2013].

Table 6 includes a comparison and evaluation of the most popular methods with respect to specific criteria based on [Bhat and Abulaish, 2015; Xie et al., 2013] and own experimental settings.

Table 6. Evaluation of graph community detection methods with respect to criteria[2].

Criteria / Method	Community Overlap	Community Hierarchy?	Outliers	Support of Weighted Edges	Support of Directed Edges	Time Complexity	Available software
OSLOM	Y	Y	Y	Y	Y	$O(n^2)$ *in worst case*	Y[3]
GANXiS (SLPA)	Y	Y	N	Y *(SLPAw)*	Y	$O(tm)$	Y[4]
MOSES	Y	N	N	N	N	$O(en^2)$	Y[5]
CFinder (CPM)	Y	Y	Y	Y *not when directed at the same time*	Y *not when weighted at the same time*	- *often polynomial*	Y[6]

Regarding the holistic approaches discussed in 2.2.1.2, we decided to evaluate them based

[2] Where n:number of nodes, k: number of communities, m:number of edges, t: a predefined max number of iterations and e: the number of edges to be expanded

[3] available at http://www.oslom.org/software.htm

[4] available at https://sites.google.com/site/communitydetectionslpa/

[5] available at https://sites.google.com/site/aaronmcdaid/downloads

[6] available at http://www.cfinder.org/

on the following criteria:

- Language independency, an important variable for social media applications,
- Overlap of Communities, because it attributes to a more realistic real-world representation,
- Directionality of relationships, as Twitter interactions are not necessarily reciprocal and it is important to be considered,
- Required Information to retrieve from Twitter, as an indicator of simplicity and easiness of data acquisition.

This evaluation is summarized in Table 7.

Table 7. Evaluation of community detection approaches with respect to criteria.

Criteria / Method	Language Independency	Overlapping Communities	Directionality	Required information from Twitter
Similarity-based				
[Zhang et al., 2012]	Partly *in textual similarity*	No	-	• Content of tweets • Following links • Retweet actions
[Beguerisse-Díaz et al., 2014]	Yes	No	Yes	Following links
[Greene et al., 2012]	Yes *Except for list selection*	Yes *(of lists & users)*	No	Twitter lists
Topology-based				
[Java et al., 2007]	Yes	Yes	Yes	Follow-type links
[Lim and Datta, 2012a, 2012b, 2013]	Yes	N/A	No	• Follower list of specific celebrities • Follow-type links (reciprocal only) • Celebrity accounts representing interest categories as seeds
[Grabowicz et al., 2012]	Yes	Yes	Yes	Follow-type links
Interaction-based				
[Correa et al., 2012]	Yes	No	Yes	• Interactions links (retweets, mentions, replies) • Topic selection
[Lim and Datta, 2012c]	Yes	Optional *possible with CPM, not Infomap*	Yes	• Follower list of specific celebrities • Mention-type interactions
[O'Callaghan et al., 2013]	Yes (method) *Expected dependency in*		No	• Follow-type links • Tweets • List memberships

	user selection			of selected accounts
Hybrid				
[Gupta et al., 2012]	No	No	No	• Tweets (content) • Follow-type links • User location of users tweeting about specific events
[Deitrick and Hu, 2013]	Mostly *Language dependency for sentiment similarity*	Optional *possible with SLPA, mot Infomap*	Yes	• Follow-type links • Interactions • Tweet content of selected accounts

The community role schemes described in 2.2.2 are evaluated and compared based on the following criteria:

- language independency, whether their inference is any dependent on language,
- role multiplicity, whether an individual can hold more than one role,
- social data type used, for example friendship-relationships, interactions etc.,
- level od detail, how many distinct roles the scheme contains,
- type of roles, the more specified description of the roles, and
- whether the scheme was defined specifically on Twitter or not.

Table 8 summarizes this evaluation.

Table 8. Evaluation of community role schemes.

Criteria / Method	Language Independency	Multiple roles per individual	Social data type used	Twitter-tailored roles	Level of detail	Type
Relationship-based						
[Guimerà and Nunes Amaral, 2005]	Yes	No	*(any social network)*	No	7	Graph- & community-based
[Scripps et al., 2007]	Yes	No	*(any social network)*	No	4	Graph- & community-based
[Fagnan et al., 2014]	Yes	No	*(any dynamic social network)*	No	6	Graph- & community-based
[Tyshchuk et al., 2013]	Yes *for community & role identification*	No	Interactions *(mention & retweet)*	Yes	3	Centrality- & Prestige-based
[Beguerisse-Díaz et al., 2014]	Yes	No	Follow-type links	Yes	5	Information flow-based
Behavior-based						
[Java et al., 2007]	Yes	No	Follow-type links	Yes	3	Posting & connecting behaviour-

						based
[Tinati et al., 2012]	Yes	Yes	Tweet metadata	Yes	5	Communicator roles
[Maulana and Tjen, 2013]	No	No	Tweets &Posts	Partly *& Facebook*	6	Goals/motivation-based
[Golder and Donath, 2004]	No	No	Conversations threads	No	6	Social roles
[Nolker and Zhou, 2005]	Yes	No	Interactions in Usenet & Discussion contribution	No	3	Contribution- and conversation-based

3 Online game behavioural analysis

SmartH2O engages users with an original mix of gaming, gamification of water consumption data, and social network activity and data analysis. Social awareness through gaming and gamification requires the user to interact with the platform explicitly, by performing some actions that are directly connected to the sustainability goals of the project. However, because users are rewarded for their action, the problem arises of monitoring the trustfulness of their actions and of avoiding malicious behaviors, a problem well studied in game design and, more generally, in disciplines addressing the analysis of users' behavior.

Recent years have seen a deluge of behavioral data from game players. The reasons for the data surge are many, including the introduction of new business models, technological innovations, and the popularity of online games. Regardless of the causes, the proliferation of behavioral data leads to the problem of how to derive and implement insights from them. Behavioral datasets can be very big, time-dependent/sensitive and high dimensional.

Game data mining is an increasingly important topic for researchers and practitioners alike. Analyzing records of in-game data provides new avenues towards understanding players and their behavior, interests, and preferences. This allows for optimizing game design, automatically adapting game contents and dynamics, economic decision making or improving player experience and game mechanics. All of this applies to different genres and platforms. Research in this area addresses questions regarding architectures and frameworks for in-game data collection and storage, algorithms and methods for in-game data mining, pattern analysis and classification as well as approaches to behavior prediction and incentive setting.

This need is particularly meaningful in games aiming not to just entertain their users but also to instruct them or to solve computational problems as byproduct of their gameplay.

As Serious Games and Games with a Purpose gain popularity and adoption, game designers look for new ways to draw players to their product or to maximize even the smallest contribution.

The subject of **player modeling in games** has been well studied over the years; however, research on player modeling is typically just applied to single player games or small-scale multiplayer games. In these studies, researchers have used player models to adapt gameplay for specific player types, generate content that more players would find satisfactory, and even discover level design mistakes during game production.

In the following, we present the requirements relevant to evaluate the contribution that player modeling could have to Serious Games; various player modeling techniques are then surveyed by outlining their strengths and weaknesses with respect to their performance for the genre, focusing on how these techniques can be used to improve player experiences through improving the design of the game.

Tracking players actions is typically easy in traditional videogames given the fact that most of them contain some form of online modes, data can be shared to the game provider's servers through the Internet, and behavioral analysis techniques can be applied to games both online and offline with no substantial changes apart from the transmission of data and the use of server or the local machine of the player.
Serious Games, on the other hand, present several challenges that do not exist in traditional games. The need of tracking the contribution of several players at the same time imposes requirements on the types of techniques that can be used to monitor and predict player behavior and possibly exclude malicious users.

By considering the many challenges existing in Serious Games adoption, a list of requirement used to evaluate player behavior monitoring and prediction techniques for traditional game design is provided; each technique is evaluated in a scale 0-3, with 0 meaning unsuitable and 3 meaning applicable to all the requirements.

Afterwards, several algorithms for real time monitoring of the contribution of the players are provided.

3.1 Requirements for online player behavioural data analysis

Six different aspects have to be considered when creating a Serious Game used to model players' behavior:

1. Scalability
2. Ability to Handle New Data
3. Authorial Burden
4. Performance on Unsupervised Tasks
5. Noise Tolerance

3.1.1 Scalability

One of the challenge that Serious Games that involve several hundreds of players at the same time is related to the fact that each of these players is producing a large amount of data with each action they perform in game. In order for a technique to be successful in such a situation, it must be able to quickly sift through a large amount of data and make predictions about future player actions in real time. Summarizing, behavioral modeling techniques must be able to quickly make predictions and must be able to be quickly trained on large amounts of player data.

3.1.2 Ability to Handle New Data

Every time a player performs any action in a Serious Game, more data is generated as a byproduct of her gameplay. The ability to efficiently incorporate this data into a learning technique of some kind is important for making accurate predictions about player behavior. If it takes a long time to incorporate new data, then it is likely that by the time new models have been developed, the data that they were built off of will be old and its use will be limited. A technique should also be able to adapt to the ever-changing environments that are common in modern Serious Games. Content is constantly being added and adapted to meet the rapid cycle of design improvements, and the last thing that a developer wants is to have to delay content release because the player modeling techniques in place cannot handle the release of this new content in an efficient manner.

3.1.3 Authorial Burden

Creating a model of player behavior can be a very difficult and very costly exercise. Creating an accurate model of player behavior can potentially be costly in multiple different ways. For example, it can be costly in terms of the time it takes for a person to come up with possible player types by hand and then exhaustively list the possible actions that each type of player could take. On the other hand, it could be a significant financial cost if multiple writers are employed to ease the time investment required to perform such a task. If one uses a computational model to describe player behavior, the creation of this model could incur a great deal of authorial burden if it requires a large amount of observational data to produce an accurate model. Since it can be difficult for some game designers to obtain large amounts of player behavior data before a game is released, this can be seen as a different, yet equally important, type of authorial burden. Ideally, a player modeling technique should minimize the amount of effort that the game's author needs to put into creating the player models.

3.1.4 Performance on Unsupervised Tasks

In machine learning and data mining, there exists the dichotomy between supervised and unsupervised learning. In a supervised learning problem, you are given training examples that are labeled with whatever behavior you want to predict. If you wanted to create a model of player behavior using a supervised learning method, for example, you would provide training examples that contained some in game behaviors that are then labeled with the player type associated with that example. In an unsupervised learning problem, training examples are not labeled, and it is up to the learner to determine how best to group examples into types. Data in Serious Games is inherently unsupervised as players cannot be defined within a player type prior of beginning to play. Even if they did, there has been work done that

calls into question the validity of self-report data [Gross et al., 1975]. The need to handle unsupervised data can be overcome either by employing an algorithm that is able to handle this type of data (such as a clustering algorithm), or somehow intelligently converting the problem into a supervised learning problem (as is typically done when manual tagging is used).

3.1.5 Noise Tolerance

One side effect of having possibly hundreds of players interacting with the system and among themselves is the fact that the game is prone on receiving noisy data. Data that is noisy is data that is difficult or impossible to interpret due to its being unstructured, being generated by a spurious source or, even worse, generated by misbehaving players, thus it is important that algorithms are able to distinguish data that contains actual predictive trends (often referred to as a predictive signal) from that which is nothing but noise. If a technique is able to do this, we say that this technique is noise tolerant.

3.1.6 Accuracy

For a player model to be useful, it must be able to accurately predict player behavior. There are many definitions of what constitute player behavior, and could include anything from predicting player actions to predicting player personality types. In Serious Games, if you are going to perform any of the tasks mentioned earlier it is of uttermost importance for your predictions to be accurate because it is oftentimes detrimental to the gameplay experience to make an incorrect prediction. This is because it could lead to tailoring content based on the assumption that this prediction is correct while it may, in reality, be wrong.

3.2 Player Behaviour Analysis Techniques

In the following, a list of the most common player behavior analysis techniques used even in commercial games is provided, defining for each of them their shortcomings and strengths and comparing it against the desiderata we have described in the previous section.

3.2.1 Manual Tagging

The act of manual tagging can be described as the act of defining a typology of players and then determining how specific actions in game reflect each individual type in this categorization. A player typology is a division of players based on some discerning criteria. Examples of this criterion include separating players by playstyle, motivations for play and skill. In order to come up with a player typology, one typically consults a domain expert and then uses insights garnered from this domain expert to discern what possible player types exist in game. This domain expert could be someone who is intimately familiar with player behavior, such as a behavioral psychologist, or even someone who is simply familiar with the genre that a particular game exists in, such as a game designer or even the author of the player models. Once this has been done, then the author must determine how every action available in the game contributes or detracts from each of the derived player types.

Despite its mechanical simplicity, this technique has remained quite popular and examples can be found in many AAA game titles. In Star Wars: The Old Republic [7] for example, Bioware uses a simple manual tagging scheme for filtering content. In this scheme, two player types exist, dark side players and light side players. While performing actions in the game, a player is given several decisions that dictate which type he or she belongs to. These decisions are manually classified into dark side and light side actions by the developers. If the player chooses to complete a quest by performing dark side actions, for example, they will probably complete the quest by using brute force methods that could endanger innocent NPCs. On the other hand, if a player chooses to complete a quest by performing light side actions they will probably be presented with content that provides a subtler, or less violent at

[7] http://www.swtor.com/

the very least, approach to complete the quest. In this scheme, Bioware drew on knowledge contained in the genre, the Star Wars universe in this case, to determine the possible player types and then manually tagged which actions were dark side actions and which actions were light side actions.

Most player typing techniques that take advantage of manual tagging follow this template. The main difference between techniques comes from where the expert knowledge is coming from. Sometimes, the expert tries to take a well-known behavioral theory and apply it to games, whereas other times the expert may simply observe gameplay and interpret how this behavior translates into discrete player types.

3.2.1.1 Manual Tagging Examples

One of the first attempts to classify players into distinct types was done by Richard Bartle [Bartle, 1996]. In this work, Bartle relies on his own observations of players in a multi- user dungeon (MUD) to determine how best to partition them. He divides players into 4 groups based on their motivations for playing:

- **Achievers**: Players that place the most value on acquiring in-game rewards and making progress in the game,
- **Explorers**: Players that place the most value on exploring the virtual world as well as exploring the capabilities of the game engine,
- **Socializers**: Players that place the most value on interacting with other players, and
- **Killers**: Players that place the most value on interfering with the gameplay of others.

Bartle also defines a set of possible actions that could be associated with each of these player types.

In 2006, Chris Bateman et al. [Bateman et al., 2006] derived a set of player types based on the Myers-Briggs typology [Myers et al., 1985]. The Myers-Briggs typology is based on a set of four dichotomies: extroversion-introversion, sensing-intuition, thinking-feeling, and judging-perceiving. A player's personality is defined through their values for each of these dichotomies. As with Richard Bartle, Bateman et al. were able to divide players into 4 distinct player types:

- **Conqueror**: These players are driven to overcome all challenges the game presents them and have other recognize them for their achievement.
- **Manager**: These players view games as a problem and seek to discover strategies and develop skills in order to solve it.
- **Wanderer**: These players are looking for a fun experience that they can use to escape their daily life, and
- **Participant**: These players want to feel like they are a member of both the game world as well as the larger game community.

Each of these types encompasses 4 of the types available in the Myers-Briggs typology.

Ryan Houlette [Houlette et al., 2004] describes a technique for creating player models that consists of creating a tree structure where the leaves represent all of the available actions that a player can take. Parents of these actions correspond to the different types of gameplay that contain these actions. For example, a player model that describes stealthy gameplay would consist of a tree and the leaves of the "stealthy gameplay" node would be actions such as uses smoke grenades and avoids guards. So, in order to use this technique, one would first have to create a set of trees to describe how each action contributes to each possible playstyle in the game.

In the PaSSAGE system [Thue et al., 2007], Thue et al. uses player models that were generated by examining Robin's guide for pen-and-paper role playing games [Robins, 2002]. In this case, Thue et al. derived a set of 5 player types from this text:

- **Fighters**: These players prefer combat and to take aggressive actions in game,
- **Power-Gamers**: These players prefer to gain special items and valuable resources,
- **Tacticians**: These players prefer to think creatively,
- **Storytellers**: These players prefer complex plots, and

- **Method Actors**: These players prefer to take dramatic actions.

Thue et al. tagged choices that the player would make in the game with the player type that would feasibly most enjoy that option. They would keep track of which types of actions the player had taken, and would use this to determine which choices to offer the player.

3.2.1.1.1 Evaluation

- **Scalability - 3**: All of the work involved in using this technique takes place during production and not actually at run time. While people are playing the game, determining how certain actions contribute to a player model is a simple lookup.
- **Ability to Incorporate New Data - 0**: If new content is generated for the Serious Game, then it must go through the same tagging process that occurred during initial game production to identify all the possible game mechanics and actions. If a substantial amount of content is added, then this task quickly becomes too cumbersome to finish in a reasonable amount of time.
- **Authorial Burden – 1:** Time must be invested to both come up with the player types in the game and to actually tag every action with these player types, with most of the time being spent during the actual tagging process. The amount of time it would be spent to tag all the content of a Serious Game can be justified just if the models are meaningful for a particular purpose, e.g. identifying players that have been able to obtain a tangible learning improvement.
- **Performance on Unsupervised Tasks – 0:** Manual tagging deals with unsupervised data by turning it into a supervised problem. The process of tagging every action with an associated player type is equivalent to adding a class label to unsupervised data.
- **Noise Tolerance - 0**: Manual tagging techniques consider all data concerning player actions to be relevant which makes it highly susceptible to noisy data.
- **Accuracy - 2**: The ability for manual tagging techniques to accurately describe player behavior depends solely on the quality of the expert knowledge that was used to tag the data. If this expert knowledge is flawed in some way, then any predictions made using these tags will also be flawed. If the knowledge is accurate, however, then it is likely that any predictions made using the tags will be accurate.

3.2.2 Collaborative Filtering

Collaborative filtering (CF) is the technique of using preferences of known users or populations to make predictions of preferences for an unknown audience. One well known application of CF is in commercial services with heavy traffic such as eBay, Amazon.com, and Netflix. For example, Netflix will make recommendations on movies to watch based on a user's viewing history. CF has also been extended to making recommendations in games to make predictions about a player's desired narrative experience [Yu et al., 2011] and to make out-of-game recommendations in Massively Multiplayer Online Role Playing Games (MMORPGs) [Li et al., 2013], [ThaiSon et al., 2013]. The collaborative filtering umbrella breaks down into two specific approaches: memory-based CF techniques and model-based CF techniques [Su et al., 2009]. Memory-based CF stores all recorded examples in memory and then will query these examples directly in order to determine preferences. Model-based CF uses recorded data as input to a machine learning algorithm in order to make a computational model of user preferences. Regardless of the approach, all major CF techniques only have access to the user's action history when making predictions. In other words, CF techniques use only the user-item data and do not use features about the users (such as their age or gender) to make predictions on their behavior [Si et al., 2013].

CF techniques can be applied for game player behavioral monitoring and prediction, by considering the actions played in the game by a user as the item-user pair on which CF techniques are applied.

3.2.2.1 Memory-Based Collaborative Filtering

Neighborhood-based CF is a common memory-based CF algorithm where the weight or similarity of two users are computed and then a prediction is made using either a simple

weighted average or a weighted average over all users compared to the target user [Sarwar et al., 2001]. The advantage of memory-based CF is its ease of implementation and performance on dense data sets, while its disadvantages include performance issues on large and sparse data sets, dependence on user ratings, and difficulty making recommendations for users that have not provided many observations for the system to use [Su et al., 2009].

Due to the issue that memory-based CF techniques have with scaling to large datasets, these methods have not seen much use in the games community. That being said, there are a few notable counterexamples. Kyong Jin Shim et al. used an algorithm called PECOTA [Silver et al., 2013] in or- der to predict performance in Everquest II[8]. The PECOTA algorithm is typically used to predict the amount of home runs that a baseball player will hit in the current year. It works by looking at the player-in-question's past performance and compares it with the past performances of every player in a corpus. It then finds nearest neighbors and uses their future performances to generate a prediction. This is the very definition of memory-based CF, except that it is used to predict home runs instead of preferences or ratings. In Everquest II, Shim et al. define performance as the time it takes to advance to the next level. This example is notable in that it used memory- based CF techniques on a large scale, MMORPG dataset; however, it is important to note that this study was performed offline since it is quite likely that it would have taken too long to be performed in a real-time setting.

Sharma et al. [Sharma et al., 2007] used memory-based CF in order to predict player preferences in an interactive narrative environment. This technique used a nearest-neighbor approach that would examine how a player advanced the story in an interactive narrative, and then determine their enjoyment of the narrative based on ratings that other players with similar story paths and ratings gave their experience.

Hingston et al. [Hingston et al., 2013] present generative techniques for mobile games. They created InfiniteWords, where players are presented with images that they need to identify. The puzzles are generated with memory-based CF.

3.2.2.1.1 Evaluation

- **Scalability – 2**: In order to make predictions, scenarios [Zook et al., 2012] and in a game that emulated the combat memory-based CF methods must first search all observed data in order to find similar users. In Serious Games, the size of this dataset will grow slowly but steadily and the amount of data that can be efficiently searched could reach its top fast enough. Data structures such as K-D trees [Wess et al., 2014] have been used to speed up this retrieval step, but the size of data can still be an issue if it is especially large.
- **Ability to Handle New Data - 3**: New data is able to be instantly incorporated since it simply has to be added to the corpus of observations that is used to make predictions.
- **Authorial Burden - 3**: Typically, the algorithm used to make these predictions only needs to be implemented once. This is usually a very simple process and is not very time consuming, meaning that it does not add much work that the designers have to do to implement it.
- **Performance on Unsupervised Tasks - 3**: The collaborative filtering problem is inherently unsupervised since it typically operates on traces of actions/ratings made by many different users. Since this data does not contain a class label and is, therefore, unsupervised, all techniques used to solve this problem must be equipped to handle unsupervised data.
- **Noise Tolerance – 1**: Techniques such as these are typically susceptible to noise; however, it is possible to modify the canonical CF algorithms in order to make them more noise resistant. One common approach to reduce the effect that noise has on predictions is to use an ensemble of many CF predictors instead of a single predictor [DeCoste et al., 2006], [Yu et al., 2012].

[8] http://www.everquest2.com/

- **Accuracy - 3**: since collaborative filtering techniques take data generated from actual users into account system used in a turn-based role-playing game [30]. This when making predictions, it is likely that the predictions made will be accurate assuming low noise.

3.2.2.2 Model Based Collaborative Filtering

Model-based approaches address the problem that memory-based CF methods have with scaling by constructing a computational model of training data in order to make predictions. Most of the time, using the model to make predictions is much faster than searching through an entire corpus of training examples, which makes most model-based CF techniques scale better than memory- based ones. This can be done with Bayes Nets [Miyahara et al., 2002], [Heckerman et al., 2001], clustering models [Chee et al., 2001] or others [Hoffman, 1999], [Shani et al., 2006]. Model-based CFs tend to perform better than memory-based CFs in large data sets [Breese et al., 1998], [Basu et al., 1998]. While model-based techniques do scale better than memory-based ones, there is an added cost up front because the models need to be trained on observation data before they can be used. This cost, however, is typically only incurred once and can be done off-line.

Zook et al. use a tensor factorization technique to predict a player's mastery of a skill in both military training [Zook et al., 2012] and in a game that emulates combat system used in a turn-based role-playing game [Zoook et al., 2013]. This technique uses a player's past performance at various skills and then predicts what their future performance will be. In this work, this knowledge was then used to generate missions that would effectively teach the user how to use a certain skill, making this type of technique very useful for an adaptive help system.

Yu and Riedl [Yu et al., 2012], [Yu et al., 2013] apply prefix-based CF to a Drama Manager which makes plot decisions in narrative games. The Drama Manager makes decisions about which plot points to include in the story and their ordering. The CF is trained by player feedback on story event ordering.

In the domain of MMORPGs, Li and Shi [Li et al.,2013] use CF to recommend items in item stores and also models the satisfaction that is associated with said item purchase. The authors use an analytic hierarchy process combined with an improved ant colony optimization technique in order to quickly converge upon possible recommendations to make.

Min et al. [Min et al., 2013] apply the model-based collaborative filtering methods of probabilistic principal component analysis (PPCA) and non-negative matrix factorization (NMF) to the domain of serious games. These techniques were used to predict student performance on learning

3.2.2.2.1 Evaluation

- **Scalability - 3**: Model-based CF techniques scale much better than memory-based ones. Making predictions using a computational model is typically a fast process that is easily scalable to hundreds of thousands of users.
- **Ability to Handle New Data - 1**: In order to incorporate new data into these models, they must be rebuilt. This can be a time consuming process; however, one typically does not need to rebuild the model until a significant amount of new data has become available. This means that, while the computational models will need to be rebuilt, they do not need to be rebuilt every time a player performs any action.
- **Authorial Burden - 3**: While model construction might take some time to complete, the training algorithms do not require very much author intervention to run. Also, model building is performed very few times. Overall, the use of these techniques requires very little effort on the part of the author in order to work properly.
- **Performance on Unsupervised Tasks - 2**: As with memory-based CF techniques, model-based techniques are very well equipped to deal with unsupervised problems. This does mean, however, that the types of models you can construct will be limited to those that can handle unsupervised data, such as clustering techniques

- **Noise Tolerance** - 2: While model-based CF techniques are more noise resistant than memory-based techniques, they are still susceptible to noisy data. There are ways to minimize this issue, however, such as ensemble learning methods.
- **Accuracy - 3**: As with memory-based CF techniques, these methods use actual user data to make their predictions. This increases the likelihood that they make accurate predictions, especially when compared to methods like manual tagging, which do not make predictions based on player observations.

3.2.3 Goal Recognition

Goal recognition is the task of reasoning about the users' intentions based on their observed actions [Kautz, 1987], [Carberry, 2001]. It assumes that the user is engaged in goal-directed behavior—that is, the user is trying to place the world into some specific state. The task of goal recognition is to predict what state the user is trying to put the world into based on the actions he has taken so far and knowledge about the domain.

Goal recognition is closely related to the problems of action recognition and plan recognition. Action recognition [Turaga et al., 2008] (also called activity recognition) is the low-level task of deciding what action a user is taking based on sensory information such as computer vision. Because a game's state is fully-observable for the developer and because game interfaces are usually semantically explicit, activity recognition is usually not needed in Serious Games. In other words, we know what the user is doing but not why [Ha et al., 2011]. Plan recognition [Kautz, 1987] is the more general and more difficult problem of predicting not only the user's final goal but also the exact plan or sequence of actions they will use to achieve it.

Goal recognition techniques can be broadly divided into two types: those based on planning systems and those based on probabilistic models. While they both accomplish the same task, they have different technological limitations.

3.2.3.1 Planning-Based Models

Early goal recognition systems (generally in the 70's and 80's such as [Kautz, 1987], [Wilensky, 1978], [Allen et al., 1980], but also as recently as 2010 [Ramirez et al., 2010]) generally used symbolic logical reasoning to deduce the user's goals. They were similar in many ways to planning systems, which construct chains of actions to explain how an agent can accomplish a goal. Planning- based systems require the designer to provide a detailed model of the domain and annotate which actions can lead to which goals, as well as the causal and temporal constraints that exist between chains of actions. As new observations of user actions are made, these systems narrow down the list of possible goals that the user might be pursuing that are consistent with the actions taken so far [Carberry, 2001]. The more "useable" an action is when planning toward some goal, the more likely that action was taken in service of that goal.

Planning-based goal recognition systems are most suitable for low-level narrative mediation. Mediation is the process of rewriting a story when the player takes actions that make the current story impossible to carry out. Both reactive and proactive narrative mediation have been studied. Reactive mediation is the process of attempting to repair a story that has been broken by the player's actions [Riedl, 2003]. Proactive mediation is the process of attempting to anticipate which player activities will break the story and find ways to prevent or support them in advance [Harriis, 2005]. Both rely on a model of goal recognition to prevent the user from breaking the current story or to incorporate the user's desired actions into the story.

While narrative mediation may be the gold standard for quests in a persistent story-driven open-world, planning-based goal recognition and narrative mediation are simply too computationally expensive to be done in real-time, even for a small number of users. Writing a planning domain to include all the constraints on all the possible actions that a player can take is too great of an authorial burden. Also, logical deductive systems like these are not very tolerant of noisy data. In short, planning-based goal recognition systems are probably not practical for use in Serious Games any time in the near future due to their inability to scale to large scenarios.

3.2.3.1.1 Evaluation

- **Scalability - 0**: Most planning-based goal recognition and narrative mediation techniques are simply too computationally expensive to be done in real time, even for a small number of users. These issues of speed can be mitigated somewhat by using faster hierarchical planners [Kautz, 1987].
- **Ability to Handle New Data - 1**: Planning-based techniques are often domain-independent and so do not change significantly when their domain models are modified. However, adding new elements to a domain model (e.g. new game mechanics or new content) often necessitates changes to existing elements.
- **Authorial Burden - 0:** The task of modeling all the actions in all the possible activities that could be performed in a Serious Game to the level of detail required by a planning system would be a massive effort, and is thus probably impractical.
- **Performance on Unsupervised Tasks - 3**: The author of the domain model must annotate which states are valid goals. While this might be considered a supervised task, it is a trivial amount of extra work given the existing authorial burden. The main strength of planning-based goal recognition systems is that they do not require a training corpus. If a domain model can be produced along with the game, it can be deployed as soon as the game is released without the need for any preliminary data collection.
- **Noise Tolerance - 0**: Techniques based on deductive logic do not handle noise well.
- **Accuracy - 2**: Many early planning-based systems were described as theories along with examples of how they could work. Most were not tested using a corpus of real-world problems, so it is difficult to gauge their accuracy. Due to their low tolerance for noise, the accuracy of planning-based techniques in Serious Games and especially GWAP is likely to be lower than desired.

3.2.3.2 Probabilistic Models

When players can pursue multiple goals in a non-linear fashion, and when they may make mistakes along the way, goal recognition is a noisy and uncertain process. For this reason, most modern techniques are based on probabilistic methods. Charniak and Goldman were some of the first to use Bayesian Networks [Charniak et al., 1993] for goal recognition, while Bui [Bui, 2003] used a variation on Hidden Markov Models to accomplish goal recognition in real time.

While these methods scale better, tolerate noise, and are potentially less onerous to the game designer, they sacrifice a level of narrative granularity. Planning-based approaches reason at the level of atomic actions and thus can mediate even the smallest part of a story. Probabilistic models require the narrative content to be broken down into individual pre-scripted chunks (e.g. scenes or chapters) which cannot be further customized and are difficult to parameterize.

The transition from plan-based models to probabilistic models happened gradually as deficiencies in early systems were addressed. One of the first advances in modern goal recognition was to replace the onerously hand-written planning domain with a corpus of plans and their associated goals. Statistical and learning models are able to infer the temporal and causal constraints on low-level actions from these corpora, when they are not explicitly provided by the author [Riedl et al., 2003]. Blaylock and Allen [Blaylock et al.,2003] used such a corpus to tune Bayes' rule to use bi-grams of observed user actions to predict what goal the user was pursuing. Their approach runs quickly (linear in the number of possible goals) and can scale to a large game. Mott, Lee, and Lester [Mott et al., 2006] used Bayesian Networks trained on a corpus of completed quests in an education game. Gold [Gold, 2010] used an Input- Output Hidden Markov Model to predict one of three high-level goals in an action/adventure game: explore, level up, or return to town. His IOHMM can be trained in real time, and it outperformed a hand-authored Finite State Machine based on expert knowledge. However, all these approaches rely on collecting a corpus of super- vised data, which may still be too great of an authorial burden given the rapid changes and the difficulty of gathering

consistent amount of unbiased users in Serious Games.

Orkin, Smith, Reckman, and Roy [Orkin et al., 2010] describe one method to reduce this burden. They collected thousands of instances of human players acting out the roles of a waiter and a diner in their online Restaurant Game. They demonstrated that a small corpus of hand-annotated game logs can be used to annotate a larger corpus automatically.

Lesh [Lesh, 1997], [Lesh, 1998] presents a recognizer-independent method for tailoring goal recognition to individual users based on their observed preferences. Gold [Gold, 2010] also demonstrated that, once a player is familiar with the game, that player's data can be used to train an Input- Output Hidden Markov Model which is more accurate for that specific player. Techniques like this can enable content which is not only adaptive based on the player's goals but also based on the player's personality and game history.

3.2.3.2.1 Evaluation

- **Scalability - 3**: Probabilistic models require time to train, but once the model is built they can run quickly, even for a large domain. Many of these models can also be arbitrarily simplified (at the cost of accuracy) if they are too slow.
- **Ability to Handle New Data - 2**: Most probabilistic models must be retained and rebuilt to incorporate new data. However, some models like Gold's [Gold, 2010] IOHMM can be updated in real time.
- **Authorial Burden - 1**: While probabilistic approaches usually do not require a detailed domain model, they still require a corpus which may be difficult to obtain and annotate.
- **Performance on Unsupervised Tasks - 1**: Even advanced probabilistic goal-recognition systems require the author to specify which states in a domain are goals. However, the relationships of actions to goals can be learned automatically.
- **Noise Tolerance - 3**: All probabilistic models can handle some degree of noise, and others can even be extended to handle complex interleaving goals.
- **Accuracy - 3**: With the shift to building models based on a corpus of real-world data came more robust evaluation metrics for those systems. Many probabilistic goal recognition systems perform well on the tasks set to them by their designers and should be adaptable to the Serious Games context. Most can be tuned to provide only high-confidence pre- dictions if those are what is desired.

4 Adversarial behaviour detection methods

Player behavioural modelling alone often is not sufficient, especially if the system aims at exploiting the contribution submitted by the consumers/players to solve computational problems. In SmartH2O several examples of this situation arise, when letting consumers:

- Input their consumption data manually.
- Provide values of psychographic variables, e.g., family composition, appliances number and types.
- Rate content and water saving recommendations.

In all these cases, one cannot rely blindly on the data submitted by the players due to several factors such as comprehension of the input to be provided or even malicious intention of the players themselves.

In this section, we analyse various techniques used to solve the adversarial behavior by redundant annotations. These techniques aggregate the annotations in different ways in order to obtain better results.

Many of these techniques are tailored to binary annotations, labelling or classification.

In literature we can identify two main classes of methodologies:

- **Non-iterative:** uses heuristics to compute a single aggregated value of each question separately [Hung et al., 2013]. Examples of these techniques are *majority voting* and *a priori quality checking*.
- **Iterative:** performs a series of iterations, each consisting of two updating steps: 1) \item updates the aggregated value of each question based on the expertise of workers who answer that question, and 2) adjusts the expertise of each worker based on the answers given by her [Hung et al., 2013]. Examples of this techniques are *expectation maximization* and *iterative learning*.

4.1 Majority voting

One of the simplest techniques used to solve the problem is majority voting. It is also known as *majority decision*.

"Majority Decision (MD) is a straightforward method that aggregates each object independently. Given an object o_i, among k received answers for o_i, we count the number of answers for each possible label l_z. The probability $P(X_i = l_z)$ of a label l_z is the percentage of its count over k; i.e. $P(X_i = l_z) = \frac{1}{k} \sum_{j=1}^{k} \mathbf{1}_{a_{\{i,j\}}=l_z}$. However, MD does not take into account the fact that workers might have different levels of expertise and it is especially problematic if most of them are spammers."

Hung et al. [Hung et al., 2013]

The majority voting method is based on mainly two assumptions:

- The number of cheaters is less than the number of good annotators.
- A great number of annotations per object are available.

The two assumptions are required to have such a high probability that the consensus of the users is equal to the right answer.

As described by Sheng et al. [Sheng et al., 2008] majority voting is mainly used in binary or classification tasks.

- The binary task consists in choosing between two possible answers YES or NO, once gathered the annotations from the users it is just required to choose the answer that has the greatest consensus among them.
- The classification task consists in choosing one class from a set of possible classes, once gathered the annotations from the users it is just required to choose the class that has the greatest consensus among them.

As explained in [Sheng et al., 2008], majority voting does perform well when the probability *p* of obtaining the right answer from a single users is greater than 50%. In this situation, the probability of obtaining the right answer using majority voting increases with the number of users, the higher is *p* the faster it tends to 100%. On the contrary, when *p* is less than 50% majority voting fails. In this situation, the probability of obtaining the right answer decreases when the number of users increases, the lower is *p* the faster it tends to 0.

This was under assumption that all the users has the same quality (probability to give a good answer). In [Sheng et al., 2008] it is even analyzed the situation of users with different quality, reaching more or less the same results.

In [Sheng et al., 2008] it is still presented an extension of the method when used in classification called "soft" labeling that obtains better results due to the multiset nature of the annotation.

Okubo et al. [Okubo et al. 2013] present a small variation of majority voting that exploits information coming from previous answers in order to assign tasks to more trustful users. After the assignment the annotations are aggregated in the exact same way as normal majority voting. Even though this version of the algorithm obtains better results it requires more knowledge related to users and the dataset, knowledge that is not always available.

Wei-Tek et al. [Wei-Tek et al. 2014] present a variation of majority voting that requires the users to communicate in order to reach a consensus before assigning the final annotation. It obtains good result when the users engage a profitable debate.

4.1.1 Evaluation

Pros

- If the assumptions are respected it generally gives good results.
- Does not require complex aggregation algorithms.
- Does not require any knowledge about the user that provided to the annotation.
- Does not require any knowledge about the dataset.

Cons

- It requires a strong assumption with respect to the number of good users.

4.2 Honeypot

The technique proposed by Lee et al. [Lee et al. 2010] and extended to the aggregation case by Hung et al. [Hung et al., 2013] is in between majority voting and a priory quality checking.

It uses a technique coming from the computer security field that is commonly used to identify malicious agents and avoid attacks.

"In principle, Honeypot (HP) operates as MD, except that untrustworthy workers are filtered in a preprocessing step. In this step, HP merges a set of trapping questions Ω (whose true answer is already known) into original questions randomly. Workers who fail to answer a specified number of trapping questions are neglected as spammers and removed. Then, the probability of a possible label assigned for each object oi is computed by MD among remaining workers. However, this approach has some disadvantages: Ω is not always available or is often constructed subjectively; i.e truthful workers might be misidentified as spammers if trapping questions are too difficult."

Hung et al. [Hung et al., 2013]

4.3 A priori quality check

Another technique used to solve the problem is to do an a priori quality check. This approach is also known as *majority voting* with gold standard or *expert label injected crowd estimation*.

"Expert Label Injected Crowd Estimation (ELICE) is an extension of HP. Similarly, ELICE also uses trapping questions Ω, but to estimate the expertise level of each worker by measuring the ratio of his answers which are identical to true answers of Ω."

Hung et al. [Hung et al., 2013]

Given the expertise level of each worker, it is possible to weight differently the different workers. This allows to filter out random annotators (not reliable) and even exploit spammers (always give the wrong answer) by negatively weighting them.

This approach generally gives better results than majority voting as demonstrated by Vuurens et al. [Smucker et al. 2011].

An example can be found in [Snow et al., 2008] where NLP tasks have been assigned to a crowd of non-experts. In this paper it has been used a gold standard coming from experts in order to evaluate the quality of the crowd.

This method allows one to obtain even better result by further analysis. It estimates the difficulty level of each question by the expected number of workers who correctly answer a specified number of the trapping questions. Finally it computes the object probability $P(X_i = l_z)$ by logistic regression that is widely applied in machine learning. In brief, ELICE considers not only the worker expertise $\alpha \in [-1, +1]$ but also the question difficulty $\beta \in [0, +1]$. The benefit is that each answer is weighted by the worker expertise and the question difficulty; and thus, the object probability $P(X_i = l_z)$ is well-adjusted. However, ELICE also has the same disadvantages about the trapping set Ω like HP as previously described. Hung et al. [Hung et al., 2013].

4.3.1 Evaluation

Pros

- Good performance.
- Robust against random and malicious annotators.

Cons

- Requires a ground-truth with a sufficient size in order to estimate correctly the goodness/expertise of the annotators.
- Requires the ability to inject the ground-truth inside the normal workflow.
- Requires a method to uniquely identify the user that has generated an annotation.
- Requires a greater number of annotations with respect to other methods, because some of them are not directly used in the aggregation, they are just used to estimate the user goodness/expertise. This requires more time, and higher costs if it is used with a paid crowdsourcing system.

Seyda et al. [Seida et al., 2011] propose a modified version of a priori quality checking that allows one to reduce the required annotations. In this version the tasks are assigned to just a subset of the crowd, this subset is identified at runtime.

4.4 Expectation maximization

Expectation maximization is an approach based on a probabilistic model, as presented by Dempster et al. [Dempster et al., 1997] and Whitehill et al. [Whitehill et al., 2009].

"The Expectation Maximization (EM) technique iteratively computes object probabilities in two steps: expectation (E) and maximization (M). In the (E) step, object probabilities are estimated by weighting the answers of workers according to the current estimates of their expertise. In the (M) step, EM re-estimates the expertise of workers based on the current probability of each object. This iteration is repeated until all object probabilities are unchanged. Briefly, EM is an iterative algorithm that aggregates many objects at the same time. Since it takes a lot of steps to reach convergence, running time is a critical issue."

Hung et al. [Hung et al., 2013]

This method outperforms a priori quality checking and is more robust to the presence of spammers as demonstrated by Vuurens et al. [Vuurens et al., 2011] and Raykar et al. [Raykar et al., 2010] even though it is sensible to the initialization. Different starting points can lead to different solutions.

4.4.1 Evaluation

Pros

- Does not require a ground-truth.
- Robust against random and malicious annotators.

Cons

- Sensible to starting point.
- Requires a method to uniquely identify the user that has generated an annotation.
- Iterative and therefore computational heavy.

A similar technique for annotator quality estimation is proposed by Ipeirotis et al. [Ipeirotis et al., 2010]. It has been tailored to multiple choice question and uses "soft" labels instead of hard ones during the estimation of both object probability and worker quality score. The score separates the intrinsic error rate from the bias of the worker, allowing for more reliable quality estimation. This also leads to more fair treatment of the workers. [Ipeirotis et al., 2010]

4.5 Iterative learning

As explained by Kerger et al. [Karger et al., 2011a] [Karger et al., 2011b] Iterative Learning is a belief-propagation-based method for annotation aggregation. As suggested by Hung et al. [Hung et al., 2013] it can be even used to estimate question difficulty.

"Iterative Learning (ITER) is an iterative technique based on standard belief propagation. It also estimates the question difficulty and the worker expertise, but slightly different in details. While others treat the reliability of all answers of one worker as a single value (i.e. worker expertise), ITER computes the reliability of each answer separately. And the difficulty level of each question is also computed individually for each worker. As a result, the expertise of each worker is estimated as the sum of the reliability of his answers weighted by the difficulty of associated questions. One advantage of ITER is that it does not depend on the initialization of model parameters (answer reliability, question difficulty). Moreover, while other techniques often assume workers must answer all questions, ITER can divide questions into different subsets and the outputs of these subsets are propagated in the end."

Hung et al. [Hung et al., 2013]

As explained in [Karger et al., 2011b] this method obtains performance similar to expectation maximization and belief propagation with a far more simple underlying model.

4.5.1 Evaluation

Pros

- Does not require a ground-truth.
- Robust against random and malicious annotators.
- Simpler model with respect to *expectation maximization* and *belief propagation*.
- Proven convergence in the binary labeling case [Karger et al., 2011b].

Cons

- Requires a method to uniquely identify the user that has generated an annotation.
- Iterative and therefore computationally heavy.

5 Integration of social awareness and consumption mining techniques for user modelling

In the preceding Sections we have reviewed the principal techniques employed in social network analysis and in game design to characterize the individual and social behaviour of users from the digital traces of their online activity.

An original aspect of SmartH2O is the **integration of the virtual and real traces** of the user's activity: the former are the virtual community activity traces discussed before, the latter the actual metered consumption data. This section discusses the integration of social network data within the traditional mining techniques applied to consumption data.

The use of the social awareness techniques has a strong potential for improving water users' models, which aim at representing the water consumption at the individual (household) level as determined by natural and socio-psychographic factors as well as by the users' response to different water demand management strategies [for a review, see Cominola et al., 2015a (under review) and references therein].

In the literature, two distinctive modelling approaches have been developed. The first one aims at the construction of **descriptive models**, which focus on the analysis of historical water consumption patterns only and provide short-term forecast of the water consumption on the basis of time series analyses [e.g., Altunkaynak et al., 2005; Alvisi et al., 2007]. Yet, these approaches neglect the social dimension of the problem, as they do not try to relate the observed consumption patterns to the socio-psychographic features of the modelled users.

An alternative approach is offered by **predictive models**, which provide estimates of the water consumption at the individual (household) level as determined by natural and socio-psychographic factors, and in response to water demand management strategies. The general formulation of a water demand predictive model for a generic user *i* is the following:

$$y_i = f(\boldsymbol{x}_i) \quad [1]$$

where y_i is the consumption profile of the *i*-th user and $\boldsymbol{x}_i$ denotes the set of *M* determinants influencing his behaviour, represented by a variety of demographic and psychographic users data (*e.g.*, age, number of house occupants, income level, conservation attitude, etc.), household attributes (*e.g.*, house size, type, garden area, etc.) and exogenous factors (*e.g.*, temperature, and precipitation, water price, etc.).

The identification of the water demand predictive model defined in Eq. [1] can be structured in the following two-step procedure:

1. ***Multivariate analysis***, which consists in the identification and selection of the most relevant inputs (i.e., natural and socio-psychographic drivers and water demand management strategies) to explain the preselected output (household water consumption);
2. ***Behavioural modelling learning***, which means model structure identification, parameter calibration and validation.

5.1 Multivariate Analysis

The multivariate analysis phase (i.e., variable selection as called in data-driven modelling) is a fundamental step to build predictive models of urban water demand variability in space and time. In most of the works, the identification of the most relevant drivers relies on the results of correlation analysis between a pre-defined set of variables (candidate drivers) and the water consumption data. Depending on the specific domains from which the candidate drivers are extracted, we can distinguish three main approaches:

- **economic-driven studies**, which focus on studying the correlation between water consumption and purely economic drivers, such as water tariff structures or water price elasticity [e.g., Olmstead and Stavins, 2009];
- **geo-spatial studies**, which assess the correlation between hydro-climatic variables and seasonality with water consumption [e.g., Polebitski and Palmer, 2010];
- **psycographic-driven studies**, which infer the influence of users' personal attributes on their water consumption, including income, family composition, lifestyle, and households' physical characteristics (e.g., number of rooms, type, presence of garden) [Matos et al., 2014].

However, in many studies the number of candidate drivers analysed is relatively small. In order to manage a large number of potentially relevant factors influencing water users' behaviours, along with their redundancy and highly nonlinear relationships, which represent major challenges for standard cross-correlation analyses, data mining techniques are employed. In particular, feature extraction techniques [Guyon and Elisseeff, 2003] can be used to identify the most relevant determinants in describing the consumption profiles of water users out of a large set of candidate drivers. On the basis of the selected determinants, a behavioural model predicting the water consumption at the household level can be identified [Cominola et al., 2015b].

Different approaches can be adopted to perform feature extraction (for more details, see D3.2 - FIRST USER BEHAVIOUR MODELS). In particular, feature extraction techniques can be classified in two main categories:

- **Feature selection**, namely algorithms that return a subset of features selected from the original dataset as the most relevant to describe the considered output variable (*i.e.*, consumption profile). This class includes, among others, the following algorithms[9]:
  - Fast Correlation Based Filter (FCBF) [Yu and Liu, 2003];
  - Correlation Feature Selection (CFS) [Zhao *et al.*, 2010];
  - Bayesian Logistic Regression (BLogReg) embedded method [Guyon *et al.*, 2002)];
  - Sparse Bayesian Multinomial Logistic Regression (SBMLR) embedded method [Cawley *et al.*, 2007].
- **Feature weighting**, namely algorithms that rank all the features according to a measure of their relevance, with no actual selection of the most relevant variables, which however are identified as the ones in the first positions of the ranking. This class includes, among others, the following algorithms:
  - CHI-square score [Liu and Setiono, 1995];
  - Information gain [Cover and Thoma, 2012].

5.2 Behavioural Model Learning

The construction of behavioural models aims at the identification, calibration, and validation of mathematical models, which describe the water consumption (i.e., output variable) as a function of the drivers identified in the multivariate analysis. In principle any data-driven model (regressors or classifiers) can be. In practice, the preferred methods (e.g., Naive Bayes Classifiers or Decision Trees) should have the following desirable features:

1. Modelling flexibility to approximate strongly non-linear functions, particularly because the relationships between the candidate inputs (selected features) and the output (consumption profile) is completely unknown a priori.

9 The 2014 version of the ASU feature selection package downloadable at http://featureselection.asu.edu/ was adopted for this study.

2. Computational efficiency to deal with potentially large data-sets, when considering large number of users.
3. Scalability with respect to the number of candidate variables to be analysed, due to the need of testing several variables with different domains and variability.

5.2.1 Individual versus multi-user models

In the behavioural modelling literature, we can identify a first class of models, named **single-user models**, which describe the consumption behaviour of individual users considered as isolated entities [e.g., Blokker et al., 2010; Cahill et al. 2013; Maggioni 2015]. These works generally rely on dynamic models or Monte Carlo techniques based on sampling of statistical distributions describing users and end-uses (e.g., number of people per household and their ages, the frequency of use, flow duration and event occurrence likelihood). Water demand patterns can be then estimated via model simulation and comparison of the results with the observed data.

A second class of behavioural models, named **multi-user models**, instead focus on studying the social interactions and influence/mimicking mechanisms among the users. The majority of these works rely on multi-agent systems, where each water user (agent) is defined as a computer system situated in some environment and capable of autonomous actions to meet its design objectives, but also able to exchange information with the neighbour agents and change its behaviour accordingly [Wooldridge 2009]. The adoption of agent-based modelling offers several advantages with respect to other approaches [see Bonabeau, 2002 and references therein]:

1. it provides a more natural description of a system, especially when it is composed of multiple, distributed, and autonomous agents;
2. it relaxes the hypothesis of homogeneity in a population of actually heterogeneous individuals;
3. it allows an explicit representation of spatial variability;
4. it captures emergent global behaviours resulting from local interactions.

As a consequence, multiagent systems can be employed to estimate market penetration of water-saving technologies [Chu et al., 2009], to simulate the feedbacks between water consumers and policy makers [Kanta and Zechman, 2014] and to study the role of social network structures and mechanisms of mutual interaction and mimicking on the behaviours of water users.

5.3 Integration of social awareness and consumption mining techniques

The water users models obtained via data mining techniques can significantly benefit from the integration with the social awareness techniques (see Figure 1) for multiple reasons.

Some agent-based models have been proposed in the last years as a first attempt to explore the effects of social networks on water conservation. Two models were for instance introduced in [Rixon et al., 2007]. In the first model, user agents are grouped in social networks. The idea is that when an agent becomes water stressed, she places peer pressure on other agents in her network to reduce water use. In the second model, a mimetic framework is used to capture the effect of imitation of water use behavior within a population of agents with different degrees of belief in water saving.

The most interesting attempt to study the role of social networks in water use behaviour is perhaps the one captured by the DAWN model [Athanasiadis et al., 2005]. There, a standard econometric model is extended with an agent-based social model describing the propagation of water conservation signals among neighbouring consumers classified according to their capability in persuading and in comprehending those signals.

In the broader setting of sustainability, in [Sissa, 2013] an agent based model that simulates how environmental awareness spreads in a system whose unsustainable consumption

should be reduced is developed.

Modelling and profiling the consumption behaviours of water users requires the availability of large and reliable datasets to ensure the statistical representativeness of the results. The use of social awareness techniques for retrieving information from the real users becomes key in order to collect such data in large-scale applications, particularly in terms of psychographic variables and estimated responses of the users to the water demand management strategies.

Moreover, social networks and graphs along with influence and trust techniques can be used for supporting the development of agent-based models and, in particular, to validate the results of the model simulations. In fact, accurately describing the single user (agent) behavior and connecting multiple users within an agent-based model does not ensure the validity of the model's results, while it is necessary to verify that the system-level properties emerged from the agent-level behaviors reproduce the observed social system.

Finally, the use of the social awareness techniques, such as the adversarial behavior detection, can contribute in matching the analysis of the observed water consumption patterns with the potential drivers generating the observed users' behaviors. This would allow validating the results of the classification of the users on the basis of their consumption and understanding if this latter is a good proxy representing different characteristics of the users.

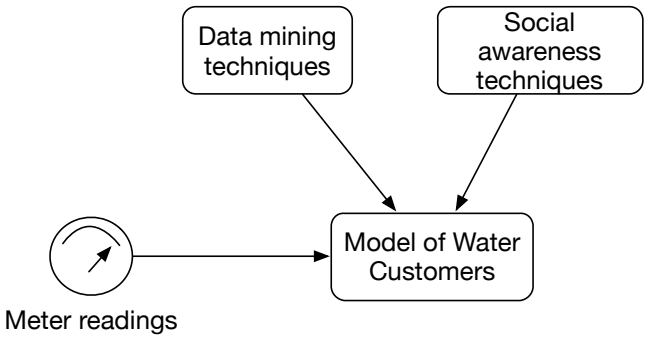

Figure 1. Integration of data mining and social awareness techniques for modeling the water users.

6 Positioning and evaluation of the proposed techniques

The preceding sections have presented a survey of the different social awareness techniques that are relevant to the goals of the SmartH2O platform.

In this Section, we provide an assessment of how social awareness techniques are positioned within the SmartH2O architecture and an evaluation of their bearing on the functionality of the affected modules.

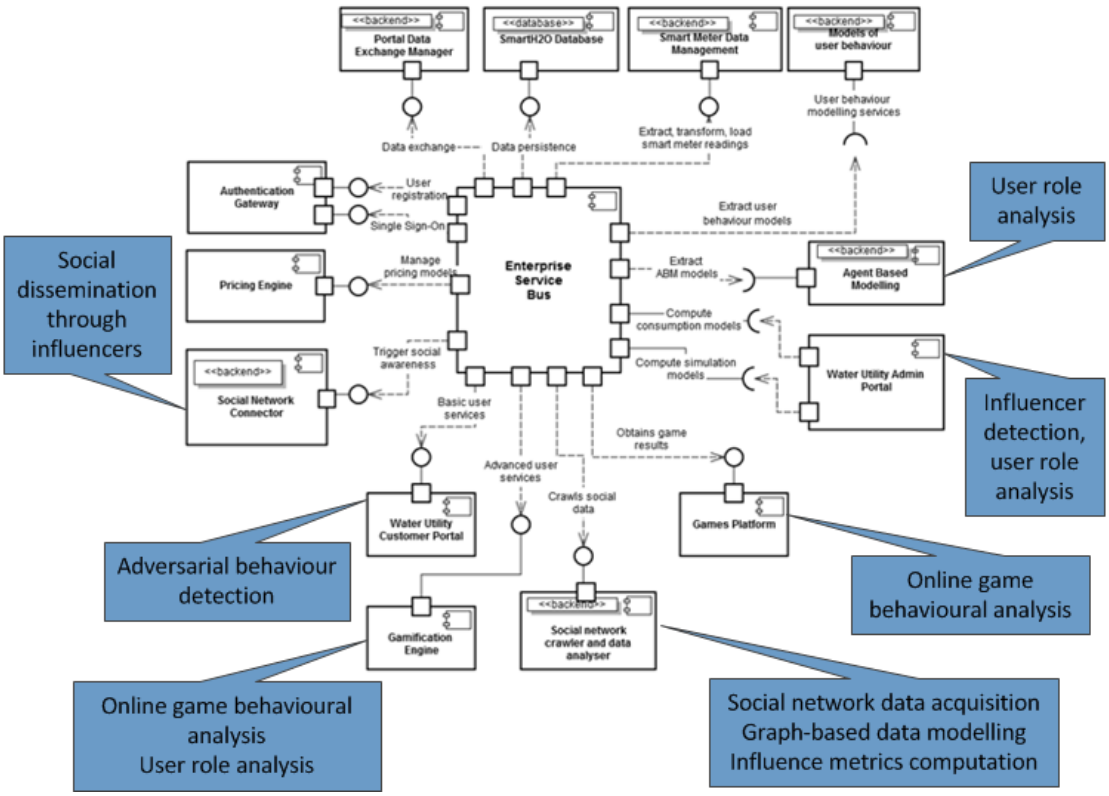

Figure 2: positioning of the social awareness techniques in the SmartH2O architecture.

Figure 2 recalls the architecture of the SmartH2O platform, introduced in D6.2 PLATFORM ARCHITECTURE AND DESIGN, and highlights which social awareness techniques are relevant to which platform modules.

The **Social Network Crawler and Data Analyser** component allows the platform to perform social data analysis to identify relevant users and content in the area of sustainable water consumption. For example, this component supports the crawling of Twitter data in order to automatically find people and content relevant for a thematic area, such as water consumption. The component will use the techniques illustrated in Sections 2.1, 2.1.4, and 2.3 in order to detect communities of interest (e.g., water saving activists) and identify influential users within them. The metrics of Section 2.3.2.1 will be evaluated, the most effective and relevant ones will be computed and a top-K, ranking-based approach will be adapted to the problem of finding influential users in the domain of sustainable water consumption.

The **Social Network Connector** component has a dual role with respect to the Social Network Crawler and Data Analyser; it allows the Consumer, Player, and Competitor users to post their achievements from the SmartH2O Water Utility Portal and Games Platform to the social network of their preference, in order to engage people from their social circle to the water consumption and sustainability campaigns of the water utility company. The module will use the results of the Social Network Crawler and Data Analyser in order to direct the social

communications of personal achievements from the water consumer towards the most relevant and influential users, so to boost the dissemination of achievement of SmartH2O users in the relevant online communities of water savers.

The **Water Utility Consumer Portal** is the component that supports the interaction between the utility customers and the SmartH2O awareness functionality. It allows the consumer to input both consumption data and values of psychographic variables; as such, it is open to spamming or erroneous data input, which can be monitored with techniques for adversarial behaviour detection, reviewed in Section 4.

The **Water Utility Admin Portal** is the component that supports the work of the supervisor in the analysis of the water consumption data and of the outcome of the gamification rules; it also supports the work of the content editor, who administers the content (e.g., tips, articles, news, etc.) published to the customers. The portal also offers interfaces to the water utility operators to run simulations, based on the models embodied in the Models of User Behaviour component and on the algorithms implemented in the Pricing Engine and in the Agent Based Modelling component. The module can benefit from the results of the Social Network Crawler and Data Analyser to direct the social communications from the utility company towards the most relevant and influential users.

The **Gamification Engine** is a back-end component that embodies rules for transforming users' actions into gamification scores and achievements. It is exploited in order to "gamify" the water consumption of the users, according to the awareness approach implemented by SmartH2O. It has an interface for the end-user, who sees the results of her water consumption actions; and administrative interfaces for the utility company's managers and operators, who can supervise the outcome of the awareness policies and define the rules that reward the actions of water consumers.

The **Games Platform** supports the execution of all the digital games of SmartH2O, including the games that are played as part of the interaction with the Drop! board game (for a description of the current status of the social awareness applications, including the SmartH2O games, see deliverable *D4.1: First social game and implicit user information techniques*. The Games Platform must also support casual players, and thus has an independent users' registration procedure, as well as a procedure for enrolling users that are already registered in the Utility Portal. The Games Platform exposes two kinds of interfaces: one or more digital games directed to the end users; an administrative interface, directed to the content editors of the game platform. The GUIs are served by a local database (the Games DB), which stores information that is pertinent only to the game play (e.g., the gaming history of players not registered in the Utility Portal). Behavioural analysis techniques will use the data stored within the Games database, aggregating and analysing them on a "per user basis" to cluster users based on different skill levels (e.g. beginners, intermediate, expert) and on their knowledge about water saving behaviours, retrieved from the proposed questions and activities, to provide tailored challenges for different kind of users.

The **Models of User Behaviour** component contains models and algorithms for profiling the behaviour of water consumers. It contains a classification algorithm that creates user segments (classes of users with similar behaviour) on the basis of their features. It also contains a disaggregation algorithm that can attribute the end uses of the total amount of water used by a household during one day, with a certain degree of approximation. This algorithm is also used to identify the relevant features to be used in classification (see D3.2 FIRST USER BEHAVIOUR MODELS). Through the use of the SmartH2O platform supplemental features will be generated, such as the influence of social awareness (obtained by the Gamification component) or the sensibility to price changes (obtained by the pricing engine).

The **Agent Based Modelling** component allows the water utility to simulate whole districts of users, thus extrapolating user models provided by the Models of User Behaviour component at a larger scale and also extrapolating the impact of network effects due to users' interactions, both in the physical and in the virtual world. The agent based model includes influence/mimicking mechanisms and social interaction among the consumers, and thus will be employed by the water utility to understand how some user types (leaders/influencers) can

stimulate a behavioural change on other users. Both components will exploit the user role analysis methods of Section 2.1.4, when SmartH2O customers are providing information on their social network accounts, to fuse the consumption and social features in a richer behavioural model.

6.1 Evaluation of the techniques with respect to language dependence

Twitter provides a huge volume of user generated content on a daily basis. This in combination with its special communication patterns, popularity in international level, speed of information diffusion and easiness of access to its data have put it in the center of a rapidly growing research field. Typical research topics are influence detection, identification of personality traits, sentiment analysis, community detection, opinion mining. Several applications rely entirely or partly on the textual content of the tweets making language an important factor.

In Twitter, around 78 different languages appear with English being the dominant one [Mocanu et al., 2013]. Considering the spatial distribution of the different languages in multilingual countries, like Belgium and Switzerland, multilingual cities as well as cities with high cultural diversity, language appears a significant variable even if an application is meant to be applied in a specific region.

Methods that depend heavily on an analysis of text use word-based n-grams and result in a huge feature space of unique words and word combinations extracted from tweets which increases the computational complexity of the dimensional space generated.

Language independency provides flexibility and universal applicability and therefore in this section we use it as an evaluation criterion of the techniques. Table 9 summarizes this evaluation.

Table 9. Techniques evaluation with respect to language independency.

	Language Independency	Detail
Influence and trust techniques	Mid to high	With the exception of centrality and activity based metrics and profile reputation (e.g., via verified accounts), influence and trust techniques are language dependent, especially when the goal includes understanding the topical context where influence is exercised.
Player Behaviour Analysis Techniques	Mid	Tagging, CF and goal recognition methods depend on the limited vocabulary used to represent classes of users and actions, so dependency is present, but to a limited scale.
Community Detection & Role analysis	High	Community Detection can be applied on a graph inferred from user interactions and/or friendship relationships. Interactions can be extracted through identifiers and tweet metadata fields. Textual content of tweets could be used only for topic/interest inference. Relationship-based roles can be inferred then trough the graph and detected communities. Behavior-based roles could be also considered based on attributes other than text (retweeting ratio etc.)
Adversarial	Low	Adversarial detection methods mostly exploit

behavior detection		Boolean or otherwise numerical or enumerative input (e.g., votes, scalar values, rankings), so language dependence is limited.

6.2 Evaluation of the techniques in small scale and large scale scenarios

In this Section, we discuss how the techniques in the categories surveyed in Sections 2, 2.4 and 4 are applicable for the SmartH2O water consumers' networks, with focus on the specificities of both the UK and ES large and uncontrolled deployment scenarios and of the Swiss smaller scale and more controlled user base.

In the evaluation of techniques we refer to two conventional scenarios, respectively small and large scale.

- A **small scale scenario** is one in which the size of the community of reference (i.e., the water consumers community) allows for an at least partially supervised approach to community detection, role analysis, influencer detection, player behavioural analysis and adversarial behaviour detection. In such a scenario, it is possible for a human operator to
  - Address communications to specific users, so to obtain missing data that could improve the quality of the user classification.
  - Validate the outcome of unsupervised algorithms manually, e.g., confirm or reject the qualification of a user as a “spammer” or “unreliable”.
  - Override or replace the outcome of unsupervised algorithms manually, e.g., assigning a new user to a specific role or consumers’ class.
- A **large scale scenario** is one in which the size of the community of reference forbids in most cases any human supervision on top of community detection, role analysis, influencer detection, player behavioural analysis and adversarial behaviour detection

Table 10. Evaluation of social awareness techniques (small scale vs large scale scenario).

Technique	Small scale scenario	Large scale scenario
Community detection & Role Analysis	Detected communities can be analysed in detail due to small size and possibility of supervising and can be monitored in time. Key role holders can play significant role in information diffusion and influence. Users that appear as outliers and in distance and expected to be relatively uninfluenced from communities’ information flow can be directly contacted.	Community Detection and role identification can be beneficial in information dissemination. Specific users with key positions within or between communities can be directly contacted providing a high likelihood of information diffusion to the communities they are attached to. Communities can be further analysed and described through their central users.
Influencer detection	Content-based metrics combined with graph metrics (on small social graphs due to the large burden and to limitations with the social	Full graph metrics are too expensive to be applied in a large scale context. Thus, influencers can be better identified with score

	network data retrieval APIs) allow one to retrieve: • Central users who propagate influence to adjacent nodes • People who mainly focus on topic-related content • People who involve other users in their communication	computation metrics that mix: • Content classification • Communication skills evaluation • Information diffusion
Online game behavioural analysis	The Major problem in small scale scenarios is due to the limited availability of data from the users, thus techniques able to handle new data and unsupervised tasks have to be preferred. Memory Based Collaborative Filtering techniques are the obvious choice here given the fact that collaborative filtering is inherently unsupervised and new data is able to be instantly incorporated, making it feasible to continuous change in the dataset corpus. Planning Based Models provide a useful alternative to most known techniques due to the fact that they do not require a training corpus even though they cannot handle noise well. Given the lack of available data, Manual Tagging techniques are not suited for the task, along with Model Based Collaborative Filtering and Probabilistic Models.	The characteristic issue in large scenario is the ability of behavioral techniques to handle data at a scale. Scaling is not a problem for manual tagging techniques, since determining how certain actions contribute to a player model is a simple lookup over the rules defined a-priori, even though they are not suitable in presence of noise and new data. Model Based techniques (Collaborative Filtering, Probabilistic) require time to train, but once the model is built they can run quickly, even for a large domain and simplified to increase speed at the cost of accuracy; the drawback is the ground truth required to train the models and which may not be immediately available for a large population of users. Memory based collaborative filtering may be feasible even at a scale but only after a certain amount of data has already been fed into the system by the users under scrutiny. Planning based models have to be discarded in scenarios requiring scale, due to their peculiar nature.
Adversarial behaviour detection	Expectation Maximization and Iterative methods require	While Expectation Maximization and Iterative

	the analysis of all the contributions at the same time. This allows one to exploit the graph structure and estimate information that cannot be inferred by local methods. While computationally intensive, these algorithms obtain high quality results even in presence of noise.	methods are scalable and can be effectively parallelized, they still require a great amount of computational power to be applied. In presence of a large amount of raw data and with a reasonably low level of noise, simpler schemes, like majority voting, can obtain acceptable results with a much lower computational cost.
Multivariate analysis and behavioural model learning	The multivariate analysis on a small number of users allows the manual validation of the outcomes of the user profiling.	Feature extraction techniques and data-driven behavioural model learning generally have good scalability with respect to the community's size, but the identification of mis-classified user profile can be difficult.

7 Conclusions and future work

In this deliverable we have provided a review of existing social network analysis, online game player behavioural analysis, trust and people search techniques, and adversarial user's behaviour detection methods employed in social games to detect malicious behaviours, such as cheating and spamming. For each technique we have proposed a general purpose assessment independent of the SmartH2O requirements, and also (in Section 6) an evaluation that considers the SmartH2O water consumers' networks, with focus on the specificities of both the UK and ES large and uncontrolled deployment scenario and of the Swiss smaller scale and more controlled user base.

In addition, in Section 5, we have established a cross reference between the research work of WP3 and WP4, discussing how the social awareness measures designed in WP4 will be exploited in the user modelling approach pursued in WP3.

After the initial setup of the social awareness approach of SmartH2O described in this deliverable, work will prosecute in the design and implementation of the social awareness functionality of the affected modules shown in Figure 2.

The **Social Network Crawler and Data Analyser** are under development for the Twitter social network and microblogging site. They implement mass scale data collection and an initial subset of the influence metrics surveyed in Section 2. Implementation will be augmented to support more refined, hybrid metrics (text- graph- and activity-based) for a more accurate profiling of influential users.

The **Social Network Connector** component will be developed in order to let users in the Water Utility Consumer Portal and Water Utility Admin Portal to post achievements and relevant content, respectively.

The **Water Utility Consumer Portal** will be augmented to include social awareness features: social sign-in, to enable the mapping of consumers to social network members, friend invitation, and posting of achievements to a social network. The requirements for such features are specified in Section 9 to 11 of D2.2 FINAL REQUIREMENTS.

The **Water Utility Admin Portal** will be extended to allow water utility admin users to post relevant content to social networks, also targeting influential users detected by the Social Network Crawler and Data Analyser.

The **Gamification Engine** will be progressively refined based on the response of the water consumers to the social and gamification stimuli. We will evaluate the opportunity of incorporating features of the user model in the reward policies, including features that characterize the social behaviour.

The **Games Platform** supports the execution of all the digital games of SmartH2O, including the games that are played as part of the interaction with the Drop! board game. Online player behavioural analysis will be applied to infer the level of expertise of players and refine the targeting of challenges to users based on their class.

The **Models of User Behaviour** will apply feature selection for the construction of the consumer’s model, including in the selection process also the features that are calculated through social network analysis, such as different influence indicators. The **Agent Based Modelling** component will be designed so to be able to include influence mechanisms and social interaction among the consumers to support the water utility to understand how some user types (leaders/influencers) can stimulate a behavioural change on other users.

8 References

[Agarwal et al., 2008] Agarwal, N., Liu, H., Tang, L., and Yu, P. S. (2008). Identifying the influential bloggers in a community. In Proceedings of the 2008 International Conference on Web Search and Data Mining, pages 207–218. ACM.

[Aggarwal et al., 2010] C. C. Aggarwal and H. Wang. Managing and mining graph data, volume 40. Springer, 2010.

[Allen et al., 1980] J. F. Allen and C. R. Perrault, “Analyzing intention in utterances,” Artificial intelligence, vol. 15, no. 3, pp. 143–178, 1980.

[Amelio and Pizzuti, 2014] Amelio, A. and Pizzuti, C. (2014). Overlapping community discovery methods: A survey. In Social Networks: Analysis and Case Studies, pages 105–125. Springer.

[Bader and Hogue, 2003] G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. BMC bioinformatics, 4(1):2, 2003.

[Barbieri et al., 2013] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. Knowledge and information systems, 37(3):555–584, 2013.

[Bartle, 1996] R. Bartle, “Hearts, clubs, diamonds, spades: Players who suit muds,” Journal of MUD research, vol. 1, no. 1, p. 19, 1996.

[Basu et al., 1998] C. Basu, H. Hirsh, W. Cohen et al., “Recommendation as classification: Using social and content-based information in recommendation,” in Proceedings of the national conference on artificial intelligence. John Wiley & Sons LTD, 1998, pp. 714–720.

[Bateman et al., 2006] C. M. Bateman and R. Boon, 21st century game design. Charles River Media Hingham, MA, 2006.

[Beguerisse-Díaz et al., 2014] Beguerisse-Díaz, M., Garduño-Hernández, G., Vangelov, B., Yaliraki, S. N., and Barahona, M. (2014). Interest communities and flow roles in directed networks: the twitter network of the uk riots. Journal of The Royal Society Interface, 11(101).

[Berger, 2001] E. Berger. Dynamic monopolies of constant size. Journal of Combinatorial Theory, Series B, 83(2):191–200, 2001.

[Bhat and Abulaish, 2015] Bhat, S. Y. and Abulaish, M. (2015). HOCTracker: Tracking the Evolution of Hierarchical and Overlapping Communities in Dynamic Social Networks. IEEE Transactions on Knowledge and Data Engineering, 27(4):1019–1013.

[Bi et al., 2014] Bi, B., Tian, Y., Sismanis, Y., Balmin, A., and Cho, J. (2014). Scalable topic-specific influence analysis on microblogs. In Proceedings of the 7th ACM international conference on Web search and data mining, pages 513–522. ACM.

[Bislimovska, 2014] B. Bislimovska. Textual and content based search in software model repositories. PhD thesis, Italy, 2014.

[Blaylock et al., 2003] N. Blaylock and J. Allen, “Corpus-based, statistical goal recognition,” in International Joint Conference on Artificial Intelligence, vol. 18 Lawrence Erlbaum Associates Ltd, 2003, pp. 1303–1308.

[Blondel et al., 2008] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008, 2008.

[Borgs et al., 2014] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 946-957. SIAM, 2014.

[Brandtzaeg and Heim, 2011] Brandtzaeg, P. B. and Heim, J. (2011). A typology of social networking sites users. Int. J. Web Based Communities, 7(1):28–51.

[Breese et al., 1998] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[Bui et al., 2003] H. H. Bui, “A general model for online probabilistic plan recognition,” in International Joint Conference on Artificial Intelligence, vol. 18. Citeseer, 2003, pp. 1309–1318.

[Carberry, 2001] S. Carberry, “Techniques for plan recognition,” User Modeling and User-Adapted Interaction, vol. 11, no. 1-2, pp. 31–48, 2001.

[Cataldi and Aufaure, 2014] Cataldi, M. and Aufaure, M.-A. (2014). The 10 million follower fallacy: audience size does not prove domain-influence on twitter. Knowledge and Information Systems, pages 1–22.

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. (2010). Measur- ing user influence in twitter: The million follower fallacy. ICWSM, 10:10–17.

[Charniak et al., 1993] E. Charniak and R. P. Goldman, “A bayesian model of plan recognition,” Artificial Intelligence, vol. 64, no. 1, pp. 53–79, 1993.

[Chee at al., 2001] S. H. S. Chee, J. Han, and K. Wang, “Rectree: An efficient collaborative filtering method,” in Data Warehousing and Knowledge Discovery. Springer, 2001, pp. 141–151.

[Chen et al., 2014] Chen, C., Gao, D., Li, W., and Hou, Y. (2014). Inferring topic-dependent influence roles of twitter users. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pages 1203–1206. ACM.

[Chen et al., 2011] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In SDM, volume 11, pages 379-390. SIAM, 2011.

[Chen et al., 2010] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’10, pages 1029-1038, New York, NY, USA, 2010. ACM.

[Chen et al., 2009] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09, pages 199–208, New York, NY, USA, 2009. ACM.

[Cheng and Church, 2000] Y. Cheng and G. M. Church. Biclustering of expression data. In Ismb, volume 8, pages 93-103, 2000.

[Clauset, 2005] Clauset, A. (2005). Finding local community structure in networks. Phys. Rev. E, 72:026132.

[Cominola et al., 2015a] Cominola, A., Giuliani, M., Piga, D., Castelletti, A., and Rizzoli, A.E. (2015). Benefits and challenges of using smart meters for advancing residential water demand modeling and management: a review. Environmental Modeling & Software (under review).

[Cominola et al., 2015b] Cominola, A., Giuliani, M., Piga, D., Castelletti, A., Rizzoli, A.E., (2015). Modeling residential water consumers’ behaviors by feature selection and feature weighting. Accepted at IAHR World Congress, 28 June-3 July, The Hague, NL.

[Cook et al., 2005] Cook, K. S., Yamagishi, T., Cheshire, C., Cooper, R., Matsuda, M., and Mashima, R. (2005). Trust building via risk taking: A cross-societal experiment. Social Psychology Quarterly, 68(2):121–142.

[Correa et al., 2012] Correa, D., Sureka, A., and Pundir, M. (2012). itop: Interaction based topic centric community discovery on twitter. In Proceedings of the 5th Ph.D. Workshop on Information and Knowledge, PIKM ’12, pages 51–58, New York, NY, USA. ACM.

[Coscia et al., 2011] Coscia, M., Giannotti, F., and Pedreschi, D. (2011). A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining: The ASA Data Science Journal, 4(5):512–546.

[DeCoste et al., 2006] D. DeCoste, “Collaborative prediction using ensembles of max- imum margin matrix factorizations,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 249–256.

[Deitrick and Hu, 2013] Deitrick, W. and Hu, W. (2013). Mutually enhancing community

detection and sentiment analysis on twitter networks. Journal of Data Analysis and Information Processing, 1(3):19–29.

[Dempster et al., 1997] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, 39(1):1_38, 1977.

[Denning, 2004] Denning, P. J. (2004). Network laws. Commun. ACM, 47(11):15–20.

[Ding, 2011] Ding, Y. (2011). Community detection: Topological vs. topical. Journal of Informetrics, 5(4):498 – 514.

[Domingos and Richardson, 2001] Domingos, P. and Richardson, M. (2001). Mining the network value of customers. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 57–66. ACM.

[Fagnan et al., 2014] Fagnan, J., Rabbany, R., Takaffoli, M., Verbeek, E., and Za¨ıane, O. (2014). Community Dynamics: Event and Role Analysis in Social Network Analysis, volume 8933, pages 85–97. Springer International Publishing.

[Farkas et al., 2007] Farkas, I., Ábel, D., Palla, G., and Vicsek, T. (2007). Weighted network modules. New Journal of Physics, 9(6):180.

[Figueiredo et al., 2011] Figueiredo, F., Benevenuto, F., and Almeida, J. M. (2011). The tube over time: characterizing popularity growth of youtube videos. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 745–754. ACM.

[Forestier et al., 2012] Forestier, M., Stavrianou, A., Velcin, J., and Zighed, D. (2012). Roles in social networks: Methodologies and research issues. Web Intelligence and Agent Systems, 0:1–17.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3-5):75–174.

[Gleave and Welser, 2009] Gleave, E. and Welser, H. (2009). A conceptual and operational definition of'social role'in online community. In 42nd International Conference on System Sciences, pages 1–11.

[Golbeck, 2009] J. Golbeck. Trust and nuanced profile similarity in online social networks. ACM Transactions on the Web (TWEB), 3(4):12, 2009.

[Golbeck, 2005] J. A. Golbeck. Computing and applying trust in web-based social networks. PhD Thesis, 2005.

[Gold, 2010] K. Gold, "Training goal recognition online from low-level inputs in an action-adventure game," in Proceedings of the 6th International Conference on Foundations of Digital Games, 2010.

[Goldenberg et al., 2001a] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing letters, 12(3):211–223, 2001.

[Goldenberg et al., 2001b] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. Academy of Marketing Science Review, 9(3):1–18, 2001.

[Golder and Donath, 2004] Golder, S. and Donath, J. (2004). Social Roles in Electronic Communities. In Association of Internet Researchers (AoIR) conference Internet Research 5.0, pages 1–25.

[Gomez Rodriguez et al., 2010] Gomez Rodriguez, M., Leskovec, J., and Krause, A. (2010). In- ferring networks of diffusion and influence. In Proceedings of the 16th ACM SIGKDD inter-national conference on Knowledge discovery and data mining, pages 1019–1028. ACM.

[Goyal et al., 2010] Goyal, A., Bonchi, F., and Lakshmanan, L. V. (2010). Learning influence probabilities in social networks. In Proceedings of the third ACM international conference on Web search and data mining, pages 241–250. ACM.

[Goyal et al., 2011] A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm

for influence maximization under the linear threshold model. In Data Mining (ICDM), 2011 IEEE 11th International Conference on, pages 211–220. IEEE, 2011.

[Grabowicz et al., 2012] Grabowicz, P. a., Ramasco, J. J., Moro, E., Pujol, J. M., and Eguiluz, V. M. (2012). Social features of online networks: The strength of intermediary ties in online social media. PLoS ONE, 7:1–14.

[Granovetter, 1978] M. Granovetter. Threshold models of collective behavior. American journal of sociology, pages 1420–1443, 1978.

[Greene et al., 2012] Greene, D., O'Callaghan, D., and Cunningham, P. (2012). Identifying topical twitter communities via user list aggregation. Proc. 2nd International Workshop on Mining Communities and People Recommenders (COMMPER 2012) at ECML 2012.

[Gross et al., 1975] S. J. Gross and C. M. Niman, "Attitude-behavior consistency: A review," Public opinion quarterly, vol. 39, no. 3, pp. 358–368, 1975.

[Guimera and Nunes Amaral, 2005] Guimera, R. and Nunes Amaral, L. A. (2005). Functional cartography of complex metabolic networks. Nature, 433(7028):895–900.

[Gupta et al., 2012] Gupta, A., Joshi, A., and Kumaraguru, P. (2012). Identifying and characterizing user communities on Twitter during crisis events. In Proceedings of the 2012 workshop on Data-driven user behavioral modelling and mining from social media - DUBMMSM '12, page 23, New York, New York, USA. ACM Press.

[Güting, 1994] R. H. Güting. Graphdb: Modeling and querying graphs in databases. In VLDB, volume 94, pages 12–15. Citeseer, 1994.

[Ha et al., 2011] E. Y. Ha, J. P. Rowe, B. W. Mott, and J. C. Lester, "Goal recognition with markov logic networks for player-adaptive games," in Seventh Artificial Intelligence and Interactive Digital Entertainment Conference, 2011.

[Hang et al., 2009] C. W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09, pages 1025-1032, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[Hannon et al., 2010] Hannon, J., Bennett, M., and Smyth, B. (2010). Recommending twitter users to follow using content and collaborative filtering approaches. In Proceedings of the fourth ACM conference on Recommender systems, pages 199–206. ACM.

[Harris et al., 2005] J. Harris and R. M. Young, "Proactive mediation in plan-based narrative environments," in Intelligent Virtual Agents. Springer, 2005, pp. 292–304.

[Huang, 2007] Huang, F. (2007). Building social trust: A human-capital approach. Jour- nal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswis- senschaft, pages 552–573.

[Huang et al., 2013] Huang, P.-Y., Liu, H.-Y., Chen, C.-H., and Cheng, P.-J. (2013). The impact of social diversity and dynamic influence propagation for identifying influencers in social networks. In Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, volume 1, pages 410–416. IEEE.

[Houlette et al., 2004] R. Houlette, "Player modeling for adaptive games," AI Game Programming Wisdom, vol. 2, pp. 557–566, 2004.

[Hung et al., 2013] Nguyen Quoc Viet Hung, NguyenThanh Tam, LamNgoc Tran, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In Xuemin Lin, Yannis Manolopoulos, Divesh Srivastava, and Guangyan Huang, editors, Web Information Systems Engineering. WISE 2013, volume 8181 of Lecture Notes in Computer Science, pages 1_15. Springer Berlin Heidelberg, 2013

[He and Singh, 2008] H. He and A. K. Singh. Graphs-at-a-time: query language and access methods for graph databases. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 405-418. ACM, 2008.

[Heckerman et al., 2001] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization,"

The Journal of Machine Learning Research, vol. 1, pp. 49–75, 2001.

[Henderson et al., 2011] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: Graph mining using recursive structural features. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pages 663–671, New York, NY, USA, 2011. ACM.

[Hingston et al., 2013] P. Hingston, C. B. Congdon, and G. Kendall, "Mobile games with intelligence: A killer application?" in Computational Intelligence in Games (CIG), 2013 IEEE Conference on. IEEE, 2013, pp. 1–7.

[Holder et al., 1994] L. B. Holder, D. J. Cook, S. Djoko, et al. Substucture discovery in the subdue system. In KDD workshop, pages 169–180, 1994.

[Hofmann et al., 1999] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in International Joint Conference on Artificial Intelligence, vol. 16. ASSOCIATES LTD, 1999, pp. 688–693.

[Ipeirotis et al., 2010] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10, pages 64_67, New York, NY, USA, 2010. ACM.

[Jabeur et al., 2012] Jabeur, L. B., Tamine, L., and Boughanem, M. (2012). Active microbloggers: identifying influencers, leaders and discussers in microblogging networks. In String Processing and Information Retrieval, pages 111–117. Springer.

[Java et al., 2007] Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pages 56–65. ACM.

[Karger et al., 2011a] David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In Advances in Neural Information Processing Systems 24 (NIPS 2011)

[Karger et al., 2011b] David R. Karger, Sewoong Oh, and Devavrat Shah. Budgetoptimal task allocation for reliable crowdsourcing systems. CoRR, abs/1110.3564, 2011.

[Kashima and Inokuchi, 2002] H. Kashima and A. Inokuchi. Kernels for graph classification. In ICDM Workshop on Active Mining, volume 2002. Citeseer, 2002.

[Kautz, 1987] H. A. Kautz, "A formal theory of plan recognition," Ph.D. dissertation, Bell Laboratories, 1987.

[Kazienko and Musial, 2007] Kazienko, P. and Musial, K. (2007). On utilising social networks to discover representatives of human communities. Int. J. Intell. Inf. Database Syst., 1(3/4):293– 310.

[Krishna et al., 2012] V. Krishna, N. Ranga Suri, and G. Athithan. Mugram: An approach for multi-labelled graph matching. In Recent Advances in Computing and Software Systems (RACSS), 2012 International Conference on, pages 19-26. IEEE, 2012.

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 137–146. ACM.

[Kiss and Bichler, 2008] Kiss, C. and Bichler, M. (2008). Identification of influ- encers—measuring influence in customer networks. Decision Support Systems, 46(1):233–253.

[Kong and Feng, 2011] Kong, S. and Feng, L. (2011). A tweet-centric approach for topic-specific author ranking in micro-blog. In Advanced Data Mining and Applications, pages 138–151. Springer.

[Kulis et al., 2009] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. Machine learning, 74(1):1–22, 2009.

[Kumpula et al., 2008] Kumpula, J. M., Kivelä, M., Kaski, K., and Saramäki, J. (2008). Sequential algorithm for fast clique percolation. Physical Review E, 78(2):026109.

[Kuter and Golbeck, 2007] U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In AAAI, volume 7, pages

1377-1382, 2007.

[Kwak et al., 2010] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In Proceedings of the 19th international conference on World wide web, pages 591–600. ACM.

[Lancichinetti and Fortunato, 2012] Lancichinetti, A. and Fortunato, S. (2012). Consensus clustering in complex networks. Scientific Reports, pages 1–7.

[Lancichinetti et al., 2011] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. PloS one, 6(4):e18961, 2011.

[Lee et al., 2010] Kyumin Lee, James Caverlee, and Steve Webb. The social honeypot project: Protecting online communities from spammers. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 1139_1140, New York, NY, USA, 2010. ACM.

[Lehmann et al., 2012] Lehmann, J., Gonçalves, B., Ramasco, J. J., and Cattuto, C. (2012). Dynamical classes of collective attention in twitter. In Proceedings of the 21st international conference on World Wide Web, pages 251–260. ACM.

[Lerman and Hogg, 2010] Lerman, K. and Hogg, T. (2010). Using a model of social dynamics to predict popularity of news. In Proceedings of the 19th international conference on World wide web, pages 621–630. ACM.

[Lesh, 1997] N. Lesh, “Adaptive goal recognition,” in International Joint Conference on Artificial Intelligence, vol. 15. Citeseer, 1997, pp. 1208–1214.

[Lesh, 1998] N. Lesh, “Scalable and adaptive goal recognition,” Ph.D. dissertation, Citeseer, 1998.

[Leskovec et al., 2007] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’07, pages 420-429, New York, NY, USA, 2007. ACM.

[Leskovec et al., 2010] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In Proceedings of the 19th International Conference on World Wide Web, WWW ’10, pages 631–640, New York, NY, USA, 2010. ACM.

[Li et al., 2013] S. Li and L. Shi, “The recommender system for virtual items in mmorpgs based on a novel collaborative filtering approach,” International Journal of Systems Science, no. ahead-of-print, pp. 1–16, 2013.

[Lian et al., 2012] Lian, J., Liu, Y., Zhang, Z.-J., Cheng, J.-J., and Xiong, F. (2012). Analysis of user’s weight in microblog network based on user influence and active degree. Journal of Electronic Science and Technology, 10(4).

[Lim and Datta, 2013] Lim, K. and Datta, A. (2013). A topological approach for detecting twitter commu- nities with common interests. In Atzmueller, M., Chin, A., Helic, D., and Hotho, A., editors, Ubiquitous Social Media Analysis, volume 8329 of Lecture Notes in Computer Science, pages 23–43. Springer Berlin Heidelberg.

[Lim and Datta, 2012a] Lim, K. H. and Datta, A. (2012a). Finding Twitter Communities with Common In- terests using Following Links of Celebrities. In Proceedings of the 3rd International Workshop on Modeling Social Media (MSM’12), pages 25–32.

[Lim and Datta, 2012b] Lim, K. H. and Datta, A. (2012b). Following the Follower: Detecting Communities with Common Interests on Twitter. In Proceedings of the 23rd ACM Conference on Hypertext and Social Media (HT’12), pages 317–318.

[Lim and Datta, 2012c] Lim, K. H. and Datta, A. (2012c). Tweets beget propinquity: Detecting highly inter- active communities on Twitter using tweeting links. In Proceedings - 2012 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2012, pages 214–221.

[Lin et al., 2012] W. Lin, X. Xiao, J. Cheng, and S. S. Bhowmick. Efficient algorithms for generalized subgraph query processing. In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 325–334. ACM, 2012.

[Liu et al., 2014] Liu, W., Pellegrini, M., and Wang, X. (2014). Detecting Communities Based on Network Topology. Scientific Reports, 4:1–7.

[Lu et al., 2012] Lu, D., Li, Q., and Liao, S. S. (2012). A graph-based action network framework to identify prestigious members through member's prestige evolution. Decision Support Systems, 53(1):44–54.

[Mahé and Vert, 2009] P. Mahé and J.P. Vert. Graph kernels based on tree patterns for molecules. Machine learning, 75(1):3–35, 2009.

[Maheswaran et al., 2007] Maheswaran, M., Tang, H. C., and Ghunaim, A. (2007). Towards a gravity-based trust model for social networking systems. In Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on, pages 24–24. IEEE.

[Mathioudakis et al., 2011] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, pages 529-537, New York, NY, USA, 2011. ACM.

[Maulana and Tjen, 2013] Maulana, A. and Tjen, S. (2013). From the Angels to the Screamers: User Segmentation and e-WOM in Social Media. International Proceedings of Economics Development & Research, 55.

[McDaid and Hurley, 2010] McDaid, A. and Hurley, N. (2010). Detecting highly overlapping communities with model-based overlapping seed expansion. In Proceedings - 2010 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2010, pages 112–119.

[Min et al., 2013] W. Min, J. P. Rowe, B. W. Mott, and J. C. Lester, "Personalizing embedded assessment sequences in narrative-centered learning environments: A collaborative filtering approach," in Artificial Intelligence in Education. Springer, 2013, pp. 369–378.

[Miyhara et al., 2002] K. Miyahara and M. J. Pazzani, "Improvement of collaborative filtering with the simple bayesian classifier," IPSJ Journal, vol. 43, no. 11, pp. 3429–3437, 2002.

[Mocanu et al., 2013] Mocanu, D., Baronchelli, A., Perra, N., Gonc ̧alves, B., Zhang, Q., and Vespignani, A. (2013). The Twitter of Babel: Mapping World Languages through Microblogging Platforms. PLoS ONE, 8(4).

[Molm et al., 2000] Molm, L. D., Takahashi, N., and Peterson, G. (2000). Risk and trust in social exchange: An experimental test of a classical proposition. American Journal of Sociology, pages 1396–1427. [Mott et al., 2006] B. Mott, S. Lee, and J. Lester, "Probabilistic goal recognition in interactive narrative environments," in Proceedings of the National Conference on Artificial Intelligence, vol. 21, no. 1, 2006, p. 187.

[Myers et al., 1985] I. B. Myers, M. H. McCaulley, and R. Most, Manual: A guide to the development and use of the Myers-Briggs Type Indicator. Consulting Psychologists Press Palo Alto, CA, 1985.

[Nolker and Zhou, 2005] Nolker, R. D. and Zhou, L. (2005). Social computing and weighting to identify member roles in online communities. In Proceedings of the 2005 IEEE/WIC/ACM International Confer- ence on Web Intelligence, WI '05, pages 87–93, Washington, DC, USA. IEEE Computer Society.

[O'Callaghan et al., 2013] O'Callaghan, D., Greene, D., Conway, M., Carthy, J., and Cunningham, P. (2013). An analysis of interactions within and between extreme right communities in social media. In Ubiquitous Social Media Analysis, volume 8329 of Lecture Notes in Computer Science, pages 88–107. Springer Berlin Heidelberg.

[Okubo et al., 2013] Okubo Yuki, Kitasuka Teruaki, and Aritsugi Masayoshi. A preliminary study of the number of votes under majority rule in crowdsourcing. Procedia Computer Science, 22(0):537 _ 543, 2013. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - {KES2013}.

[Orkin et al., 2010] J. Orkin, T. Smith, H. Reckman, and D. Roy, "Semi-automatic task recognition for interactive narratives with eat & run," in Proceedings of the third Intelligent Narrative Technologies Workshop. ACM, 2010.

[Otte and Rousseau, 2002] Otte, Evelien; Rousseau, Ronald (2002). "Social network

analysis: a powerful strategy, also for the information sciences". Journal of Information Science 28: 441–453. doi:10.1177/016555150202800601

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerankncitation ranking: Bringing order to the web.

[Pal and Counts, 2011] Pal, A. and Counts, S. (2011). Identifying topical authorities in microblogs. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 45–54. ACM.

[Palla et al., 2005] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435:814–818.

[Papadopoulos et al., 2012] Papadopoulos, S., Kompatsiaris, Y., Vakali, A., and Spyridonos, P. (2012). Com- munity detection in social media. Data Mining and Knowledge Discovery, 24(3):515–554.

[Peleg, 1997] D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. In Proc. of 3rd Colloquium on Structural Information and Communication Complexity, pages 152-169, 1997.

[Quercia et al., 2011] Quercia, D., Ellis, J., Capra, L., and Crowcroft, J. (2011). In the mood for being influential on twitter. In Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom), pages 307–314. IEEE.

[Ramirez et al, 2010] M. Ramirez and H. Geffner, "Probabilistic plan recognition using off-the-shelf classical planners," in Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI2010), 2010.

[Raykar et al., 2010] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. J. Mach. Learn. Res., 11:1297_1322, August 2010.

[Riedl et al., 2003] M. Riedl, C. J. Saretto, and R. M. Young, "Managing interaction between users and agents in a multi-agent storytelling environ- ment," in Proceedings of the second international joint conference on Autonomous agents and multiagent systems. ACM, 2003, pp. 741–748.

[Robin, 2002] R. D. Laws, Robin's Laws of good game mastering. Steve Jackson Games, 2002.

[Romero et al., 2011] Romero, D. M., Galuba, W., Asur, S., and Huberman, B. A. (2011). Influence and passivity in social media. In Machine learning and knowledge discovery in databases, pages 18–33. Springer.

[Rosvall and Bergstrom, 2008] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences of the United States of America, 105(4):1118–1123.

[Saez-Trumper et al., 2012] Saez-Trumper, D., Comarela, G., Almeida, V., Baeza-Yates, R., and Benevenuto, F. (2012). Finding trendsetters in information networks. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1014–1022. ACM.

[Saigo et al., 2008] H. Saigo, N. Krämer, and K. Tsuda. Partial least squares regression for graph mining. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 578-586, New York, NY, USA, 2008. ACM.

[Saigo et al., 2009] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda. gboost: a mathematical programming approach to graph classification and regression. Machine Learning, 75(1):69–89, 2009.

[Sarwar et al., 2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.

[Satuluri and Parthasarathy, 2009] V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: Applications to community discovery. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pages 737-746, New York, NY, USA, 2009. ACM.

[Scripps et al., 2007] Scripps, J., Tan, P.-N., and Esfahanian, A.-H. (2007). Node roles and community structure in networks. In Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, WebKDD/SNA-KDD '07, pages 26–35, New York, NY, USA. ACM.

[Scripps et al., 2009] Scripps, J., Tan, P.-N., and Esfahanian, A.-H. (2009). Measuring the effects of preprocessing decisions and network forces in dynamic network analysis. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 747–756. ACM.

[Seida et al., 2011] Ertekin .eyda, Hirsh Haym, and Rudin Cynthia. Approximating the

wisdom of the crowd. In Proceedings of the Second Workshop on Computational Social Science and the Wisdom of Crowds (NIPS 2011), 2011.

[Shani et al., 2006] G. Shani, D. Heckerman, and R. I. Brafman, "An mdp-based recommender system," Journal of Machine Learning Research, vol. 6, no. 2, p. 1265, 2006.

[Sharma et al., 2007] M. Sharma, M. Mehta, S. Ontano´ n, and A. Ram, "Player modeling

evaluation for interactive fiction," in Proceedings of the AIIDE 2007 Workshop on Optimizing Player Satisfaction, 2007, pp. 19–24.

[Shetty and Adibi, 2005] Shetty, J. and Adibi, J. (2005). Discovering important nodes through graph entropy the case of enron email database. In Proceedings of the 3rd international workshop on Link discovery, pages 74–81. ACM.

[Sheng et al., 2008] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 614622, New York, NY, USA, 2008. ACM.

[Si et al., 2003] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in Machine Learning International Conference, vol. 20, no. 2, 2003, p. 704.

[Silver, 2003] N. Silver, "Introducing pecota," Baseball Prospectus, vol. 2003, pp.

507–514, 2003.

[Smucker et al., 2011] Mark D. Smucker. Crowdsourcing with a crowd of one and other trec2011 crowdsourcing and web track experiments. In In Proceedings of the Text REtrieval Conference (TREC), 2011.

[Snow et al., 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast, but is it good?: Evaluating non-expert annotations for natural language tasks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 254_263, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[Su et al., 2009] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, p. 4, 2009.

[Sun and Ng, 2013] Sun, B. and Ng, V. T. (2013). Identifying influential users by their postings in social networks. In Ubiquitous Social Media Analysis, pages 128–151. Springer.

[Tan et al., 2010] Tan, C., Tang, J., Sun, J., Lin, Q., and Wang, F. (2010). Social action tracking via noise tolerant time-varying factor graphs. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1049–1058. ACM.

[ThaiSon et al., 2013] N. ThaiSon and L. Siemon, "Impact of sequence mining on web- page recommendations in an access-log-driven recommender system." from Labelers of Unknown Expertise, page 2035_2043. December 2009.

[Thue et al., 2007] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: A player modelling approach," in Artificial Intelligence and Interactive Digital

Entertainment conference, Stanford, CA, 2007, pp. 43–48.

[Tian and Patel, 2008] Y. Tian and J. M. Patel. Tale: A tool for approximate large graph matching. In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pages 963-972. IEEE, 2008.

[Tinati et al., 2012] Tinati, R., Carr, L., Hall, W., and Bentwood, J. (2012). Identifying communicator roles in twitter. In Proceedings of the 21st International Conference Companion on World Wide Web, WWW '12 Companion, pages 1161–1168, New York, NY, USA. ACM.

[Turaga et al., 2008] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 18, no. 11, pp. 1473–1488, 2008.

[Tyshchuk et al., 2013] Tyshchuk, Y., Li, H., Ji, H., and Wallace, W. A. (2013). Evolution of communities on twitter and the role of their leaders during emergencies. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '13, pages 727–733, New York, NY, USA. ACM.

[Vuurens et al., 2009] Jeroen Vuurens, Arjen P. De Vries, and Carsten Eickho_. How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy. In Matthew Lease, Vaughn Hester, Alexander Sorokin, and Emine Yilmaz, editors, Proceedings of the ACM SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR 2011), pages 48_55, Beijing, China, July 2011.

[Wasserman and Faust, 1994] Wasserman, S. and Faust, K. (1994). Social Network Analysis: Methods and Applications, volume 8.

[Weitzel et al., 2012] Weitzel, L., Quaresma, P., and de Oliveira, J. P. M. (2012). Measuring node importance on twitter microblogging. In Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, page 11. ACM.

[Welch et al., 2011] Welch, M. J., Schonfeld, U., He, D., and Cho, J. (2011). Topical semantics of twitter links. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, pages 327–336, New York, NY, USA. ACM.

[Welser et al., 2007] Welser, H. T., Gleave, E., and Fisher, D. (2007). Visualizing the Signatures of Social Roles in Online Discussion Groups. Journal of Social Structure, 8(2):1–32.

[Weng et al., 2010] Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). Twitterrank: finding topic-sensitive influential twitterers. In Proceedings of the third ACM international conference on Web search and data mining, pages 261–270. ACM.

[Wei-Tek et al., 2014] Tsai Wei-Tek, Wu Wenjun, and M.N. Huhns. Cloud-based software crowdsourcing. Internet Computing, IEEE, 18(3):78_83, May 2014.

[Wess et al., 1994] S. Wess, K.-D. Althoff, and G. Derwand, "Using k-d trees to improve the retrieval step in case-based reasoning," Topics in Case- Based Reasoning, pp. 167–181, 1994.

[Whitehill et al., 2009] Jacob Whitehill, Paul Ruvolo, Ting fan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels

[Wilensky, 1978] R. Wilensky, "Why john married mary: Understanding stories involving recurring goals," Cognitive Science, vol. 2, no. 3, pp. 235–266, 1978.

[Xie et al., 2013] Xie, J., Kelley, S., and Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Comput. Surv., 45(4):43:1–43:35.

[Xie and Szymanski, 2012] Xie, J. and Szymanski, B. (2012). Towards linear time overlapping community detection in social networks. In Tan, P.-N., Chawla, S., Ho, C., and Bailey, J., editors, Advances in Knowledge Discovery and Data Mining, volume 7302 of Lecture Notes in Computer Science, pages 25–36. Springer Berlin Heidelberg.

[Xie et al., 2011] Xie, J., Szymanski, B. K., and Liu, X. (2011). SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In

Proceedings - IEEE International Conference on Data Mining, ICDM, pages 344–349.

[Yan and Han, 2002] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In Data Mining, 2002. ICDM Proceedings. 2002 IEEE International Conference on, pages 721–724. IEEE, 2002.

[Yan et al., 2007] X. Yan, M. R. Mehan, Y. Huang, M. S. Waterman, S. Y. Philip, and X. J. Zhou. A graph-based approach to systematically reconstruct human transcriptional regulatory modules. Bioinformatics, 23(13):i577–i586, 2007.

[Yang and Leskovec, 2010] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In Data Mining (ICDM), 2010 IEEE 10th International Conference on, pages 599-608. IEEE, 2010.

[Yang and Leskovec, 2015] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 42(1):181–213, 2015.

[You et al., 2006] C. H. You, L. B. Holder, and D. J. Cook. Application of graph-based data mining to metabolic pathways. In Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on, pages 169–173. IEEE, 2006.

[Young, 2006] H. P. Young. The diffusion of innovations in social networks. The Economy As an Evolving Complex System III: Current Perspectives and Future Directions, 267, 2006.

[Yuan and Mitra, 2013] D. Yuan and P. Mitra. Lindex: a lattice-based index for graph databases. The VLDB Journal, 22(2):229–252, 2013.

[Yu et al., 2002] K. Yu, X. Xu, J. Tao, M. Ester, and H.-P. Kriegel, “Instance selection techniques for memory-based collaborative filtering,” in Proc. Second SIAM Intl Conf. Data Mining (SDM02), 2002.

[Yu et al., 2012] H. Yu and M. O. Riedl, “A sequential recommendation approach for interactive personalized story generation,” in Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 71–78.

[Yu et al., 2013] H. Yu and M. O. Riedl, “Data-driven personalized drama management,” in Proceedings of the 9th AAAI Conference on Artificial Intelligence in Interactive Digital Entertainment, 2013.

[Zafarani et al., 2014] Zafarani, R., Abbasi, M., and Liu, H. (2014). Social Media Mining: An Introduction. Cambridge University Press.

[Zhai et al., 2014] Zhai, Y., Li, X., Chen, J., Fan, X., and Cheung, W. K. (2014). A novel topical authority-based microblog ranking. In Web Technologies and Applications, pages 105– 116. Springer.

[Zhang et al., 2012] Zhang, Y., Wu, Y., and Yang, Q. (2012). Community Discovery in Twitter Based on User Interests. Journal of Computational Information Systems, 3(February):991–1000.

[Zhong et al., 2013] M. Zhong, M. Liu, Z. Bao, X. Li, and T. Qian. Mvp index: Towards efficient known-item search on large graphs. In Database Systems for Advanced Applications, pages 193-200. Springer, 2013.

[Zhou et al., 2009] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. Proc. VLDB Endow., 2(1):718–729, Aug. 2009.

[Zook et al., 2012] A. Zook, S. Lee-Urban, M. Drinkwater, and M. Riedl, “Skill-based mission generation: A data-driven temporal player modeling approach,” in Proceedings of the 7th International Conference on Foundations of Digital Games, 2012.

[Zook et al.,2013] A. E. Zook and M. O. Riedl, “A temporal data-driven player model for dynamic difficulty adjustment,” in Proceedings of the Conference on Artificial Intelligence and Interactive Digital Entertainment. AAAI (2012, to appear), 2012.