# Modeling Player Self-Representation in Multiplayer Online Games using Social Network Data

by

Chong-U Lim

B.Eng. Computing, Imperial College London (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

© Chong-U Lim, MMXIII. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Electrical Engineering and Computer Science
May 22, 2013

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
D. Fox Harrell
Associate Professor of Digital Media
Computer Science and Artificial Intelligence Laboratory
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

# Modeling Player Self-Representation in Multiplayer Online Games using Social Network Data

by

Chong-U Lim

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

## Abstract

Game players express values related to self-expression through various means such as avatar customization, gameplay style, and interactions with other players. Multiplayer online games are now often integrated with social networks that provide social contexts in which player-to-player interactions take place, such as conversation and trading of virtual items. Building upon a theoretical framework based in machine learning and cognitive science, I present results from a novel approach to modeling and analyzing player values in terms of both preferences in avatar customization and patterns in social network use. To facilitate this work, I developed the *Steam-Player-Preference Analyzer (Steam-PPA)* system, which performs advanced data collection on publicly available social networking profile information. The primary contribution of this thesis is the *AIR Toolkit Status Performance Classifier (AIR-SPC)*, which uses machine learning techniques including k-means clustering, natural language processing (NLP), and support vector machines (SVM) to perform inference on the data. As an initial case study, I use *Steam-PPA* to collect gameplay and avatar customization information from players in the popular, and commercially successful, multi-player first-person-shooter game *Team Fortress 2* (TF2). Next, I use *AIR-SPC* to analyze the information from profiles on the social network *Steam*. The upshot is that I use social networking information to predict the likelihood of players customizing their profile in several ways associated with the monetary values of their avatars. In this manner I have developed a computational model of aspects of players' digital social identity capable of predicting specific values in terms of preferences exhibited within a virtual game-world.

Thesis Supervisor: D. Fox Harrell
Title: Associate Professor of Digital Media
Computer Science and Artificial Intelligence Laboratory

# Acknowledgments

I would like to take this opportunity to thank my supervisor, Professor Fox Harrell for all his support, help and guidance throughout the entire process of undertaking the work for this thesis. I sincerely appreciate his continuous effort in helping to align my interests with my research, and for generously imparting his knowledge and expertise in the various, cross-disciplinary fields in order to undertake this research. Thank you for your confidence and belief throughout this process.

I extend my love and thanks to my ever-supportive and loving parents who have shown endless support in all of my decisions—from my interest in computer science to entering graduate school. All that I have achieved would not have been possible without the love and care that you have shown.

A big thank you to my loving brother, my biggest inspiration, who has always been a source of knowledge and support for everything ranging from computer science to Team Fortress 2.

Finally, to my loving girlfriend Adeline Bay who is my source of happiness and motivation to press on each day. Thank you for always being there for me, and for keeping me sane throughout the challenges and difficulties. None of this would be possible without you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Values related to self-expression are often seen as subjective matters associated with issues such as individual preference and emotional disposition. They often represent individual internalization of more widespread social norms. For example, people dress in certain ways to reflect their individual opinions about fashion, thereby expressing an agglomeration of social and personal knowledge regarding clothing. However, when taking a large number of people into consideration, distinctive categories may become apparent. For example, in fashion we can distinguish between the categories of "formal," "business casual," or "leisurewear." At the same time, one person might categorize a particular outfit as business casual while another person, perhaps raised in a posh environment, sees it as mere leisurewear. Due to the subjective nature of such issues, it is not obvious how such values-laden categorization phenomena may be identified or modeled algorithmically. This thesis focuses on specific types of values associated with self-expression in social networks and online video games, with implications for self-expression of social identity at large.

## 1.1 Motivation

The field of artificial intelligence (AI) in computer science has shown significant progress and relatively successful results in areas related to computational categorization and classification tasks [32]. However, there is still much work to be done in developing more robust systems that can better model aspects of cognitive categorization [49]. Existing research has shown success with using an interdisciplinary approach to model cognitive categorization computationally [13]. I aim to build upon and extend upon such methods further to address more complex models of cognitive categorization (which I describe in Section 2.1.2). A computational approach to such phenomena has the potential to allow us to better understand issues from other domains, for example, social scientific issues relating to discrimination and (unwarranted) stigmatization [26]. This forms the basis for developing technologies that can help people better understand the significance of such issues and ensure that developers and users have the capabilities to prevent them.

## 1.2 Project Overview

Using the commercial online multiplayer game *Team Fortress 2* by *Valve Corporation*[1] (described in detail in Section 2.3.1 of the Theoretical Framework chapter), I have designed and implemented two systems to extract and identify features associated with how players choose to represent their virtual characters, along with attributes describing aspects of their social network formation. The first system, the *Steam Player-Preference Analyzer (Steam-PPA)*, performs advanced data collection and analysis on these features. The second system is the *AIR Status Performance Classifier (AIR-SPC)*, and is the primary contribution of this thesis. Using a combination of AI machine learning methods, it constructs high-level abstract features in order to highlight the important dimensions which formulate a user's social network structures, termed **meta-ties**.

A key portion of this thesis focuses on modeling preferences by player's through their avatar customization and performance in the virtual environment implemented in *Team Fortress 2* using these meta-ties. The preferences are modeled computationally using what D. Fox Harrell and I have called **status performance** [35], which encompasses virtual items with real-world monetary values. The notion of "status"

[1] Official website: `http://www.valvesoftware.com/`

comes from the exhibited behavior of players who use such items to express themselves within the game, which games studies scholar Christopher describes as “symbols of status, authority, or dominion [45].” I wish to use this model to better understand the relationship between how users choose to represent themselves using computational digital technologies, and how aspects of their real-world identity influences these decisions.

## 1.3 Contributions

In this section, I outline the following contributions that the work in this thesis has made. The contributions are:

1. The *Steam-PPA* system, which enables the collection of publicly available data from players on *Steam*, through the *Steam Web API* and from the *Steam Community Pages*. The *Steam-PPA* provides a common interface to access the informatino.

2. I present the notion of **status performance** in games as avatar customization through equipping/collecting of virtual items, each with community-derived real-world monetary value. It forms a quantitative measure of aspects of players’ preferences as “status symbols”. The AI k-means clustering algorithm is used to identify categories of players, based on their status performance, in an unsupervised manner.

3. I extend work on predicting social connections called “tie strength” [23] in social networks by extracting tie strength measures in the games-oriented social network *Steam*. This is the first time that the *Steam* network has been analyzed in this way, to the best of my knowledge. This analysis is performed using the *AIR-SPC* system. It uses natural language processing (NLP) for sentence/-word segmentation, and performs sentiment analysis for classifying the words according to conveyed emotions.

4. I demonstrate the effectiveness of dimensionality reduction using Principal Component Analysis (PCA) over feature vectors of tie strength predictive variables, aggregated using the *AIR-SPC system*. This results in a smaller set of features, termed **meta-ties**, which still sufficiently describes the whole dataset, while providing an abstract, but human understandable, way to reason about the distribution of the players.

5. Combining the results from calculating status performance and the meta-ties of players, I present results exhibiting a correlation between the two relatively separate domains of the social network and the game. I highlight the ability to use AI learning and classification techniques (Support Vector Machines) to predict a player's gameplay preference (status performance) using the player's social networking information (meta-ties).

## 1.4 Related Work

This thesis draws upon several research areas, including sociology, cognitive science, and game studies, along with AI and machine learning techniques for clustering, natural language processing, supervized learning and classification. A brief account of important references used follows.

In [45], Moore has provided, to the best of my knowledge, the first scholarly account of the various aspects of TF2 such as virtual items and achievements, and their implications for players' real-world identities. I share Moore's motivation for studying **hats** in TF2 as artifacts expressing players' preferences, and he argues that they form "achievements, but not representation of skills," and that "meaning is routed through the absurdist quality of the games' melange of historical, philosophical and popular pastiche, individual taste and expression." AI researcher Hugo Liu has argued that social networking profiles often comprise taste statements, which can be used to define a user's **taste performance** [37]. Taste performance can be understood as a computational instance of what sociologist Erving Goffman classically termed everyday self-presentation or performance [20]. I extend upon the notion of taste performance by considering the performance of avatar or player controlled game characters. These avatars (or game characters) can be viewed as what James Gee calls "projected identities" that incorporate elements of both real and the virtual identities [17]. When real-world cultural ideas of the human player are projected onto the avatar, the result is a type of blended identity that Harrell terms a "phantasmal identity" due to its blend of sensory imagery with concepts drawn from particular worldviews regarding social categories [29]. Harrell has argued that current computational identity systems are limited in their abilities to adequately represent the "dynamic contingency of real life identity experiences" [26]. Toward better addressing this gap, Harrell, in his Advanced Identity Representation (AIR) Project, introduced the cognitively-grounded AIR model for developing identity representation technologies, which aims to overcome such limitations [28] by enabling dynamic, cross-domain user self- representations (e.g., between social networks and games). An outcome of research in the AIR Project is the continuing development of the AIR Software Toolkit, of which *AIR-SPC* is a part, to support more robust and dynamic forms of user/avatar categorization and users' deployment of multiple self-representations for different purposes [30].

The *Steam* network was analyzed by computer scientists Roi Becker, et al., [2] who highlighted, among other results, that the "friendship ties", or number of friends per user, correlated with activity on the network. Their definition of "ties" differs from mine. In the work presented in this thesis, I use a more formal definition of measuring the relationship between people, based on several factors defining a player's social network, termed **tie-strength** by sociologist Mark Granovetter [23]. He outlined the importance of considering weak ties (e.g., acquaintances) for "discussion of relations between groups" and for analyzing "segments of social structure not easily defined in terms of primary groups". Eric Gilbert and Karrie Karahalios have identified ways to predict tie strength in social media [19], while Ferrera et al., point out the roles of both strong and weak ties in the Facebook social network [11]. These illustrate that social networks are appropriate systems to analyze and understand features of a person's real-world identity and social structures. Machine learning clustering and classification applications in multiplayer games have been performed by AI researchers

Anders Drachen, et al., who used k-means clustering and Simplex Volume Maximization (SiVM) on high-dimensional telemetric data (e.g., playing time, scores, kill/death ratio) to categorize players according to behaviors [10]. Additionally, performing classification using k-means clustering and Support Vector Machines (SVM) have been used for dynamic difficulty adjustments for shooter-type games [38] by computer scientists Marlos Machado, et. al. Similar clustering and classification approaches was performed for automatic preference modeling of virtual agents in strategy games by games researchers Ruck Thawonmas and Masayoshi Kurashige [55]. These outline the effectiveness of AI clustering and classification techniques for performing inference of player behavior in multiplayer online games.

## 1.5 Thesis Outline

This content of thesis is structured as follows:

- Chapter 1, this chapter, has introduced the work undertaken in this thesis, together with providing motivation, summarizing contributions, and presenting a brief account of related work.

- Chapter 2 presents the theoretical framework that this work is based upon. In the the chapter, I first provide the background information regarding the disciplines that this work draws upon. Next, I present the theoretical grounding of the formulation of the research problem. Following which, I provides details about the target application domains that this work focuses on. Lastly, I describe how it has been applied in the development of approaches and systems as part of this research.

- Chapter 3 covers the implementation details about both the *Steam-PPA* and *AIR-SPC* systems. First, I presents the experimental design for obtaining the dataset of player profiles and performing k-means clustering and principal component analysis to obtain the features describing their social network termed “meta-ties’. The chapter covers algorithmic (machine learning) approaches for further clustering analysis, model selection, model training and performing classification of status performance using these meta-ties.

- Chapter 4 presents the results from the data collection, and the results from tie strength parameter estimation for player profiles on *Steam*. I present the results from the k-means clustering analysis and process in determining the ideal number of status performance categories describing the dataset. I also cover the resultant meta-ties and their descriptions as a result of performing principal component analysis. Finally, the chapter covers the results from the training, model selection and classification performance using support vector machines.

- Chapter 5 presents analyses of the data obtained, from both the use of quantitative methods and qualitative reflection.

- Chapter 6 presents a discussion on the potential implications on the findings of this research, and concludes with a reflective discussion of the results that has been achieved, their implications, and avenues for building upon them in future work.

# Chapter 2

# Theoretical Framework

In this chapter, I present the underlying theoretical concepts and models driving the work and research in this thesis. In Section 2.1, I present the background information about the various disciplines and research areas that this thesis draws upon. In Section 2.2, I present the key concepts from Harrell's Advanced Identity Representation (AIR) Project [26] (to which this thesis contributes), which highlights the limitations of currently existing computational technologies for identity representation. The AIR project presents the cognitive-grounded AIR model as an approach to developing more robust and dynamic systems that aim to overcome such limitations. In Section 2.3, I cover the application domains of *Steam* and *Team Fortress 2* that the work in this thesis has been applied to. In Section 2.4, I outline the correspondences between the theoretical framework and the approach used in the designing and implementation of the systems in this thesis.

## 2.1 Theoretical Background

In this section, I go into detail about the concepts and theoretical background about the various disciplines and research areas that this thesis draws upon. Firstly, I describe the area of computer-supported cooperative work (CSCW) and how it is motivated by the need to address social issues that are currently inadequately supported by computational technologies. Secondly, I introduce theories and topics from the field of cognitive science, in particular the topic of cognitive categorization. Thirdly, I provide an overview of computational models of classification and present discussion of how they relate to cognitive categorization.

### 2.1.1 Computer-Supported Cooperative Work

There are aspects of society that have the potential of being improved with the effective use computers. However, it would be erroneous to believe that the technology, along with its improvements, is itself solely sufficient to transform society, i.e., the belief called "technological determinism" [22]. There is a symbiotic relationship between technology and society, and it is a complex network of relationships involving humans and technologies as situated in social context that determines whether the technology has a positive or negative effect. As such, it is important to consider factors including both the society and its technologies when analyzing the contributions that both make toward improvements. For example, computational models based on large bodies of data collected have resulted in the development of computational models that aim to support making inferences that previously were difficult for humans to judge or were not feasible to automate. To cite a more specific case, predicting the likelihood of a patient suffering a relapse based on symptoms exhibited can help healthcare institutes to more effectively serve patients' needs. Examples like this relate to the field of Computer-Supported Cooperative Work (CSCW) [24]. CSCW research findings have revealed that, due to the nature and nuances of human activity, computational systems need to reflect the nuances of highly contextualized activities in forms such as information transfer, roles, and policy.

#### The Social-Technical Gap

There are certain social problems that we are aware of, but are unsure of how to support computationally. This division in knowledge is what Mark Ackerman terms as the social-technical gap [1]. He presents numerous findings over the past 20 years, highlighting the significance of the social-technical gap and the importance of addressing it. An example is the Platform for Privacy Preferences Project (P3P) of the World Wide Web Consortium (W3C). Difficulties exist due to differences between the social aims of giving users the choice and control over their private information and the technical challenges of fulfilling those aims computationally. The differing goals between the users of the system and the institutions or companies that run them also add towards the social-technical gap. The work in this thesis highlights the close relationship between the a user's real-world social structures and his or her computational representation within the digital environments of a social network and a videogame. Consequently, it motivates the need for designers of such systems to consider and realize the importance of providing adequate capabilities in such technologies.

### 2.1.2 Cognitive Categorization

Historical approaches to understanding and describing categorization of objects in the world have been called "classical" theories (or "folk" theories when discussing everyday nonacademic categorization) [34]. According to such approaches, a category is defined by a set of characteristics, called defining features, which are necessary and sufficient conditions for identifying if objects are members of that category. Thus, categories are mentally represented as definitions, or as sets of logical predicates. For example, in the realm of geometrical shapes, we may treat *number of points* as a defining feature, or as the predicates: `points(triangle, 3)`, `points(square, 4—`, `points(pentagon, 5)`, . . . , and so on.

However, there are several limitations to the classical theories. First, based on empirical psychological experiments, Eleanor Rosch observed that a category might possess members, which are "better examples" or more representative of the category. Classical theories are inadequate to support this observation since under such theories all members of a category possess the same attributes as one another within a category, so none could be more or less representative than the other. Second, categorization is affected by human neurophysiology, body movement, and the human capacity to form mental images, all of which classical theories also fail to describe adequately. These limitations of classical theory have been overcome in newer paradigms, such as the cognitive linguistics approach to categorization, including both *prototype theory* and *exemplars theory*, both commonly considered together as the *typicality view*. Furthermore, cognitive linguist George Lakoff's work extends on these prototype effects, suggesting that human categorization is performed by using both human experience (i.e. perception, motor activity) and imagination (metaphor, metonymy, mental imagery) [34]. These different approaches reveal a paradigm shift in the research field relating to cognitive categorization.

### Typicality View

**Prototype Theory** Prototype theory, presented by Rosch [51], was a challenge to classical approaches to categorization. It presented solutions to problems related to categorization encountered by the classical theory [50, 52]. In this theory, there is the concept of typicality that can be described by a partial ordering of members, with some being more "typical" than others by possessing many features common with the prototypical member of the category than to a member of a different category. It affirms that categories can be defined by prototypes representing the typical characteristics of objects of a category, rather than necessary and sufficient conditions. The theory theorizes that people tend to identify categories of objects based on prototypical members. Next, reasoning about a category's members is performed by referring to a precise typical object in the category called a **prototype**. A prototype may be a maximally typical actual example, possessing the most number of common features, or may be a summary representation, termed an "ideal", of what a most typical example would be.

**Exemplars Theory** A related point of view of categories consists of considering them as a collection of stored exemplars in memory. This theory, known as exemplar theory, was proposed for the first time by psychologists Douglas Medin and Marguerite Schaffer in 1978 [43]. Categories are represented collectively by a set of known positive members of the category, termed the **exemplars**. The biggest difference between the exemplars theory and the others rests in the rejection of the idea that humans may have a single member, or single set of traits, that is able to describe the whole category.

There are many extensions of either theories, and even some which aim to combine or borrow aspects of one another in order to provide a more robust conceptual framework for adequately explain phenomena regarding cognitive categorization and overcome inadequacies present in each individual theory. Psychologists Gregory Murphy, et al., discuss the idea of conceptual coherence [48] in an attempt to explain that conceptual categories are coherent to the extent that they fit people's background knowledge or naive theories about the world.

#### Metonymic Models

More recently, Lakoff has proposed that conceptual categories form "idealized cognitive models (ICMs) upon which categories of objects in the world are built [34]. They are governed by four structuring principles – propositional structure, image-schematic structure, metaphoric mappings, and metonymic mappings. One of the key ideas of cognitive categorization is the phenomenon of using an aspect of something in place of the object as a whole (or a different aspect). Psychologists Barbara Tversky and Kathleen Hemenway suggest that conceptual **metonymy** forms the basis of cognitive categorization, which is in turn based on structures in physical experience [56]. Lakoff describes the example of one waitress saying to another, "The *ham sandwich* just spilled beer all over himself", wherein the term *ham sandwich* is used to stand for, and represent, the customer eating the sandwich. This illustrates the concept of metonymic models, which refer to the ICMs that contain such stands-for relations. These "prototype" effects were identified by Lakoff. In [26], Harrell relates the "prototype" effects to social identity and computational identity phenomena, and this is presented in Table 2.1.

### 2.1.3 Computational Models of Categorization

Having outlined the relevant theories of cognitive categorization and their importance, I shall now present potential ways to model such phenomena computationally. Mur-

| Type | Description |
|---|---|
| Representatives/Prototypes | Most typical or “best example” members of categories |
| Stereotypes | Normal, but often misleading, category expectations: (e.g., gender stereotypical categories define normative expectations for language use) |
| Ideals | Culturally valued categories even if not typically encountered |
| Paragons | Defining categories in terms of individual members who represent either an ideal or its opposite |
| Salient Examples | Memorable examples used to understand/create categories |

Table 2.1: Several Types of Metonymic Models and their Descriptions

phy [47] suggests that the goal of modeling categories is through the understanding of the representations of categories that we build and the means by which we perform different cognitive tasks (e.g., recognition of new objects, inferences, communication, etc.) In computer science, the field of AI treats knowledge representation as an active area of research and discussion [16]. This is particularly relevant with attempts to link the AI systems toward cognitively grounded theories.

Even though there have been discussions about the limitations of representing concepts and categories in AI systems [12], the research area of machine learning is particularly relevant with its focus on developing computational classification systems based on various learning, search, and optimization principles.

### Instance-based learning

In both the prototype and exemplar theories, the emphasis is on representative instances within categories. These representative instances are then used to define the categories and help to determine a previously unseen instance's membership. In the field of machine learning, a computational system resembling such a description is that of **instance based learning**. The system makes use of several instances stored within a knowledge-base. Classification of a new instance is determined by performing inference based on the saved instances, and assigns it an appropriate label. This classification corresponds to prototype theory when the saved instances are used to form an abstract, representative member – a prototype; whereas it corresponds to exemplar theory when a subset of the saved instances within each category are used to collectively classify the previously unseen instance. When the instances addressed require more complex data structures, the common term used is **case-based reasoning**. The parallels between such machine learning systems and the field of cognitive categorization make it a natural starting point to begin exploring more complex computational models of cognitive categorization [7, 8, 14].

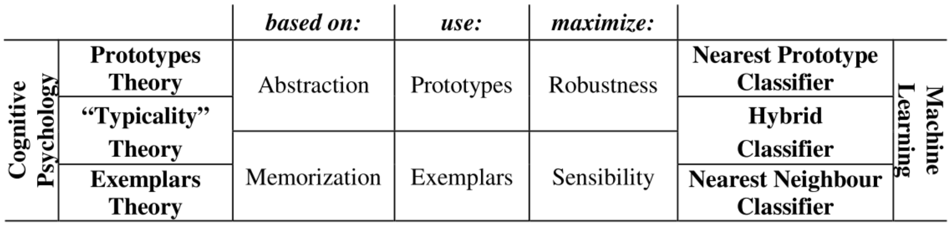
### Classification

In performing classification, besides defining measures based on performance of the classification in terms of accuracy, it is worthwhile to define some measures about the instances or categories used to perform the classification. This enables one to relate the performance and characteristics of the classifiers towards categorization theory. Francesco Gagliardi defines two measures, the **robustness** and **sensibility** of the systems as:

- Robustness: The robustness of these classifier systems can be understood by observing that the presence of outliers (due to noise or atypical instances of the class) has little or no influence on barycenter calculation, defined as the point which lies equidistant between two or more category clusters.

- Sensibility: The sensibility of these classifier systems is due to a sort of "data fidelity." The classification of new instances is merely based on the comparison between new instances and the previous observed ones. This occurs without any distinction between noisy or correct observations (also described as typical and atypical data points).

Gagliardi illustrates the relationships between the classifiers using an evaluation method which measures cognitive categorization theory based on prototypes and exemplars. The instance-based classifiers, Nearest Prototype Classifier (NPC) and Nearest Neighbor Classifier (NNC), each correspond to the limit cases of maximum robustness and maximum sensibility respectively and their relationship to the theories of prototypes and of exemplars are shown in summarized in Figure 2-1.



| Cognitive Psychology | | *based on:* | *use:* | *maximize:* | | Machine Learning |
|---|---|---|---|---|---|---|
| | **Prototypes Theory** | Abstraction | Prototypes | Robustness | **Nearest Prototype Classifier** | |
| | **"Typicality" Theory** | | | | **Hybrid Classifier** | |
| | **Exemplars Theory** | Memorization | Exemplars | Sensibility | **Nearest Neighbour Classifier** | |

Figure 2-1: Cognitive Categorization & related Machine Learning Classifier Systems (Gagliardi, 2009 [15])

**Prototype-based Classifiers** A related approach, defined as the Nearest Multiple-Prototype Classifier (NMPC) [3], works by creating abstract representative instances (prototypes) as the centroids of a subset of the represented category . Such systems obtain robust classifications, being insensitive to outliers or noise.

**Exemplar-based Classifiers** The Nearest Neighbor Classifier (NNC) and its generalized form, the k-Nearest Neighbors Classifier (k-NNC), make use of all instances in the knowledge-base and forms a subset for the representative instances. There is no extrapolation or abstraction involved, only observed instances. These systems thus fair better with *sensibility* than robustness, since outliers and noise would be also considered as part of the representative instances when performing classification.

**Hybrid Classifiers** Hybrid classifiers use both prototype-based and exemplar–based classifiers in order to balance the trade-off between sensibility and robustness. Gagliardi developed a classifier system, termed the Prototype-Exemplar Learning Classifier (PEL-C) [13], which encompasses aspects of both prototype and exemplar theories by defining abstract representative members of a category based on characteristics of both theories. The learning phase uses concepts from the NPC, while the classification is performed by a NN-like rule system. The success of such hybrid classifiers makes the case for exploring more ways to combine and extend upon current systems with the aim of developing more expressive, robust, and sensible systems.

### Genetic Algorithms

There has been some success in applying genetic algorithms (GAs) for the detection of prototypes [54]. In the *Off Broadway* system, a GA was used to identify prototypical instances in the case-base before using nearest-neighbors for classification. Ludmila Kuncheva and James Bezdek showed that GAs performed well in prototype selection for nearest prototype classification (NPC) [33]. These results highlight the potential for effectively using machine learning techniques like GAs, perhaps together with learning systems, in the cognitive categorization domain. I plan to investigate and extend upon the use of GAs to be able to model Lakoff's conceptual metaphor– and metonymy– based concepts of cognitive categorization.

## 2.1.4 Computational Digital Identity Systems

Here, I present several systems and application areas that represent users using digital identity systems. I discuss about the ways in which users' identity is computationally represented. I also outline potential limitations with current technologies in providing support for users to adequately represent themselves within the systems.

### Social Networks

Social networking profiles provide various means to define one's digital identity with various system affordances, which are often defined explicitly by system designers or group owners. As an example of potential improvements to the limitations which current social networks possess, social networking profiles could avoid limiting users to a pre-determined set of choices due its system design or underlying computational representation of its users. Based on Liu's findings that social networking profiles have information which may be used to define user taste performances [37], Harrell, and his student Greg Vargas, illustrated an approach, using the AIR model, on how user taste and implicit categories could be defined based on such taste metrics and network ties [30, 58].

### Videogames, Virtual Worlds and Interactive Narratives

The next application area in which digital representation of identity comes into play is in the realm of digital entertainment, particularly videogames. In videogames, as defined by Gee [17] and elaborated by Harrell [28], users construct computational identities by projecting aspects of their real-world identities onto virtual representations. Games researchers Doris Rusch and Matthew Weise suggest that games, virtual worlds, or interactive narratives can allow the study and critical analysis of identity and social phenomena within the medium [53]. AI and games researchers Josh McCoy, et al., used AI-driven authoring systems for the modeling of social interactions between people virtually [41, 42] in the game "Prom Week". The interactions by users within virtual environments have been shown to impact and reflect social issues from their real-world users, particularly when involving negative or socially unacceptable actions [9, 46, 61]. This outlines both its power for effect and the importance of understanding more about the development of systems in order to prevent social issues such as stigmatization and stereotyping from occurring. There has been a great deal of research into the use of machine learning and classification techniques in videogames for behavior or player-modeling [18, 10]. My aim is to adapt and develop more technologies to extend upon the uses of such techniques by aligning them with models of cognitive categorization. It should be possible to use digital media to explore the effectiveness and results of such AI systems.

## 2.2 Advanced Identity Representation (AIR)

The construction of one's identity is closely linked both to one's means of representation of himself or herself and to his or her social categories. This manifests itself in the physical world through avenues like behavior, physical appearance, speech, and language. These representations are not static, but may be dynamic across different contexts such as social situations. Code-switching is the phenomenon that may occur when 1) a multilingual individual substitutes a word or phrase from one language with a word or phrase from another [31], or 2) an individual makes use of different vernacular patterns within the same language. It is an example of representing one's identity differently across multiple groups by using different styles of language, or speech, with different groups of people.

Additionally, computational identity representations exhibit different nuanced characteristics. These manifest themselves in online virtual environments such as multiplayer video games, social networks, forums, and blogs; often in the form of an online alias, handle, or a dynamic multimodal representation such as an avatar or game character. Nick Bostrom defines these as digital identities, which are digital representations of real-world identities that link a number of attributes [4]. However, such computational media within which digital identities reside have various problems related to both the technical limitations and the social issues which are created because of the lack of sufficiently robust frameworks to adequately support the nuances in identity [21, 5, 28].

### 2.2.1 Computational Components of Identity Representation

In representing one's self digitally in the form of avatars, game characters, and profiles, several norms for behavior and group affiliations are established that may introduce problems such as prejudices, stigmatization, stereotypes and other associated social issues. Harrell's National Science Foundation supported project "Computing for Advanced Identity Representation" (the AIR Project) [26] constitutes an interdisciplinary approach to the design of technologies for identity representation by enabling imaginative self-representations and implementing dynamic social identity models grounded in computer science and cognitive science. Harrell motivates the need for better technologies to provide more support against such social problems through more robust and dynamic systems that also raise critical awareness about the problems in the infrastructures of existing technologies.

Harrell describes the components that are most commonly used across the various computational identity technologies as the shared technical underpinnings [25]. These are shown in Figure 2-2. By observing the cross-platform correspondences between the components listed, we can begin to address current limitations of existing systems more holistically across computational identity technologies (e.g., avatars, profiles, accounts, and player characters) and move towards the construction of more robust and flexible systems that are able to adequately deal with the highly contextualized and nuanced aspects of social identity within digital environments [17, 27].

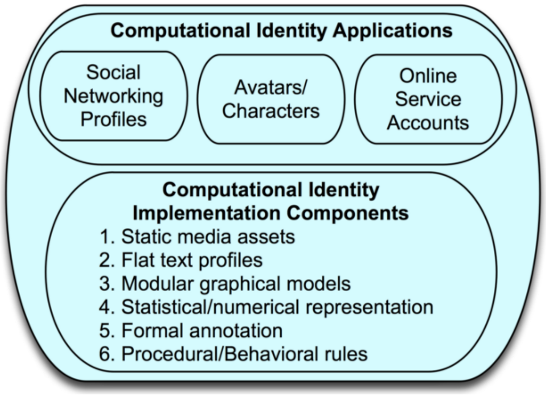

Computational Identity Applications
Social Networking Profiles
Avatars/ Characters
Online Service Accounts
Computational Identity Implementation Components
1. Static media assets
2. Flat text profiles
3. Modular graphical models
4. Statistical/numerical representation
5. Formal annotation
6. Procedural/Behavioral rules

Figure 2-2: Shared Technical Underpinnings of Computational Identity Applications (Harrell, 2009) [25]

### 2.2.2 The AIR Model

The AIR Model leverages cognitive approaches to categorization from cognitive linguistics and sociological approaches to classification from science studies. These were covered earlier in Section 2.1. Additionally, it outlines arts/humanities-based strategies for addressing identity phenomena informed by fields including semiotics, cultural studies, and art theory. Figure 2-3 summarises how the cognitive building blocks of identity, such as conceptual metaphor and prototypes, are the basis for building social identity representations, affordances for identity performance, and subsequently computational identities [26]. Using such a model allows us to align our understanding of computational structures together with our understanding of how users model categories as imaginative cognitive processes. We may then use these understandings to construct computational systems backed by theories of cognitive categorization.

With this interdisciplinary approach and framework, the aim is to develop better computational techniques, systems and applications that address and improve upon the problems inherent in many current systems.

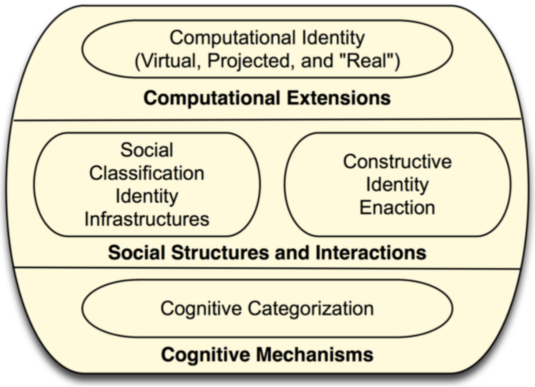


Figure 2-3: The AIR Model of Cognitively Grounded Computational Identity (Harrell, 2009) [26]

## 2.3 Application Domains

In this section, I provide the background information for both of my target domains. Firstly, I present the the multiplayer online game *Team Fortress 2* (TF2). I outline the basic features of the game, along with its capabilities and structures that support user self-representation through avatar customization. Secondly, I cover the social network and distribution platform *Steam*, and outline its prevalence as a social networking platform, together with its infrastructures and features which supports player identity representation within their network.

### 2.3.1 Team Fortress 2

*Team Fortress 2*[1] (TF2) is an online multiplayer first-person-shooter (FPS) videogame developed by Valve Corporation and released in 2007 as a sequel to its predecessor *Team Fortress Classic*. Since then, it has been received more than 300 updates, which

[1] Official Website: `http://www.teamfortress.com`

Figure 2-4: The Official Promotional Image of Team Fortress 2 (Valve Corporation)

have included entirely new gameplay modes, maps, features and items (in addition to bug fixes and improvements.)[2]

### Gameplay

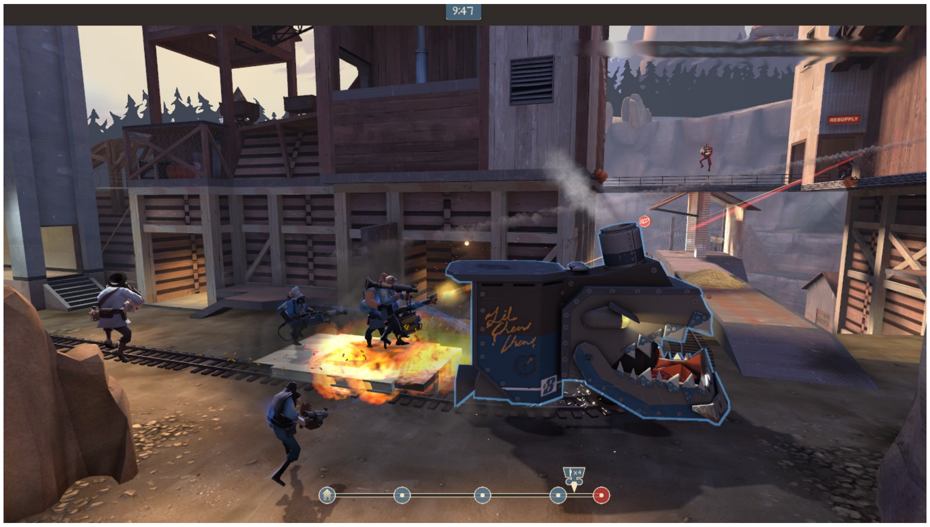


Figure 2-5: In-game screenshots of two teams in battle. Team BLU is pushing the detonation device, while Team RED is trying to prevent further progress.

The game pits players from two opposing teams (**RED** versus **BLU**) one another

[2] The latest version, as of writing, is v1.2.4.9, released on 1 February 2013.

in various game modes. There are total of seven game modes, each with corresponding types of maps that form the environment in which the players engage in battle. For example, the *payload* game mode has maps prefixed with the pl_ tag and involves one team having to transfer a detonation device from one point in the map to the other while the opposing team works to prevent it. Another popular game mode is *Capture the Flag* (CTF) (maps prefixed with ctf_), in which both teams have to seek out, steal, and bring back an opposing team's "intelligence," TF2's version of a flag (an item which both teams possess and must protect from being stolen by the opposing team).

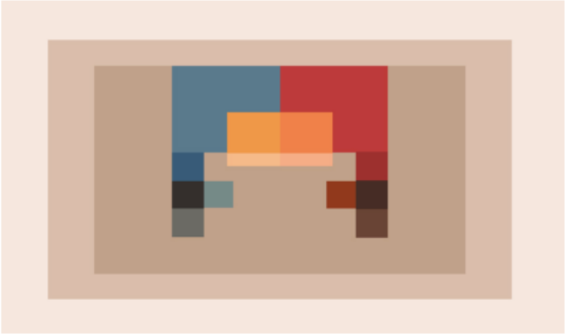


Figure 2-6: Color scheme for the opposing red and blue teams [44]

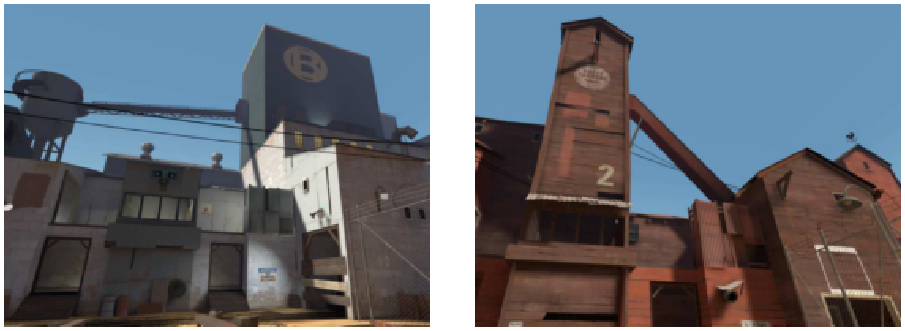
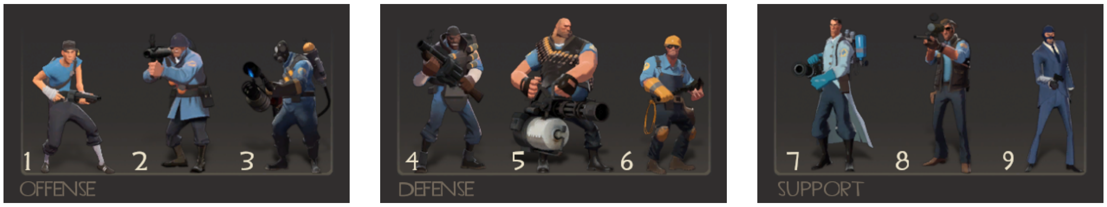


Figure 2-7: In-game screenshots of buildings belonging to the two teams within the map environment in TF2 [44]

Players in each team are visually differentiated with a color scheme (Figure 2-6). The color scheme is used in the each game to visually mark characters, objects, and buildings corresponding to each team (Figure 2-7).

### Character Classes and Roles



(a) Offensive classes (b) Defensive classes (c) Support classes

Figure 2-8: The 9 Character Classes in Team Fortress 2, grouped by their roles. [57]

Players in TF2 choose to play as a character from one of nine available character classes. Each character class has a unique visual 3-dimensional (3D) model and has different attributes and abilities, as well as different weapons. Each team is often composed of players playing as different classes, as each class has strength and weaknesses, which teams need to balance out in order to be more effective at accomplishing their goals. The classes can be classified into three distinct **roles**:

1. **Offensive**: Consisting of the *Scout*, *Soldier* and *Pyro* (Figure 2-8(a)). These are the main attacking classes due to attributes which make them more adept for capturing the "Intelligence" flags (i.e., the extremely fast movement speeds which Scouts possess).

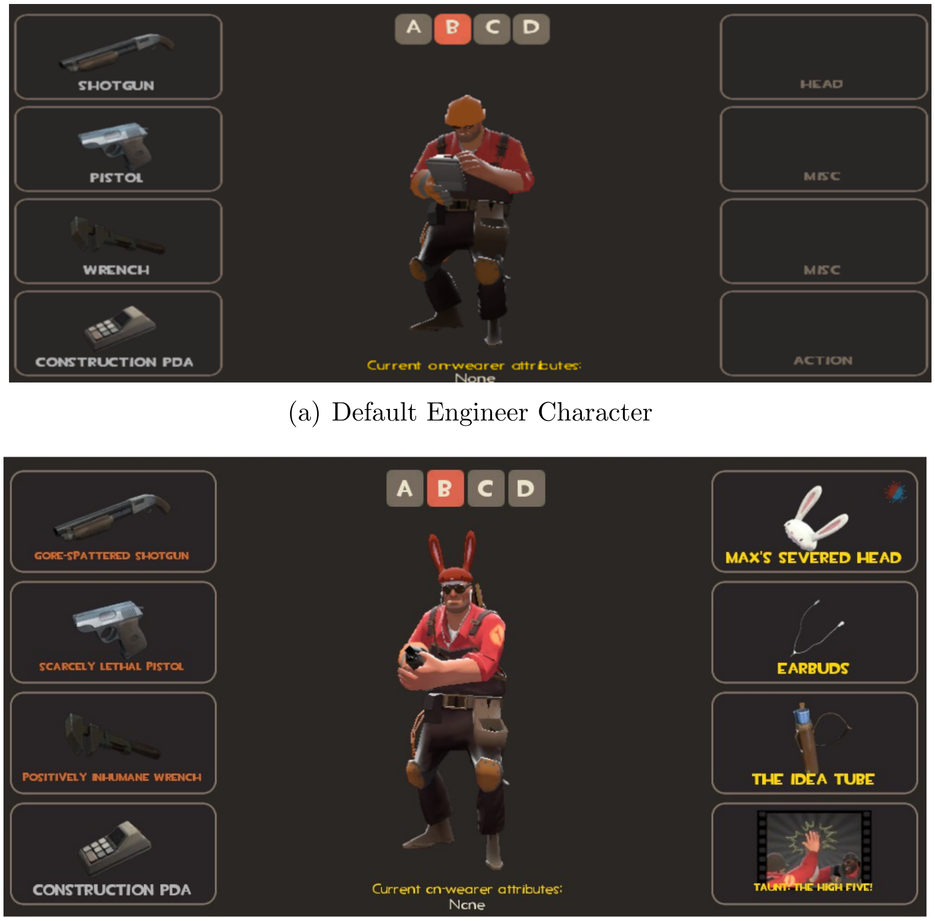
2. **Defensive**: Consisting of the *Demoman*, *Heavy* and *Engineer* (Figure 2-8(b)). These classes serve to prevent opponents from accomplishing their goals (e.g., stalling the progress of opponents who are moving the detonation device or preventing the "Intelligence" from being stolen). They possess the most firepower of the groups [57]. Their attributes also reflect upon their roles, e.g., the Heavy class has the highest health-points[3] of all the classes.

3. **Support**: Consisting of the *Medic*, *Sniper* and *Spy* (Figure 2-8(c)). These classes often possess specialized abilities and attributes that make them neither good attackers nor defenders, but when deployed alongside players of the other

[3] Health-points (or hit-points) correspond to a higher amount of damage a character may be subjected to before getting killed.

classes, are extremely effective at giving their teams an advantage. As examples, the *Medic* possesses a healing device which restores the health points of their team-mates and the *Spy* has the ability to turn invisible or even take the identity of an opponent player (assuming the enemy player's name and visual appearance) in order to confuse opponents.

### Character Customization



Current on-wearer attributes:
None

(a) Default Engineer Character

Current on-wearer attributes:
None

(b) Customized Engineer Character

Figure 2-9: Customizing Characters in TF2. The

Each game character and class has a base visual appearance and a set of associated attributes. However, players are able to customize their characters through a *Loadout* menu (Figure 2-9). There are 8 customizable *slots*, ranging from the *primary weapon*, *secondary weapon* and to *headwear*. By default, classes have no headwear or accessories, and are equipped only with a default set of weapons and equipment (Figure 2-9(a)). Players are able to customize their characters with items that provide functional benefits (e.g., weapons that deal more damage) and items that modify the visual appearance of the character (e.g., a whimsical hat with rabbit ears), both of which are shown in Figure 2-9(b).

There are a variety of ways in which these items may be obtained, such as: purchasing them with real-world currency, receiving them as rewards granted for accomplishing tasks, or finding them as randomly “dropped” items while playing the game. Due to the various acquisition methods, some items, particularly some hats, are deemed more valuable, and players engage in community-based exchange and trading of these virtual items in order to obtain some of the rarer or more valuable ones.

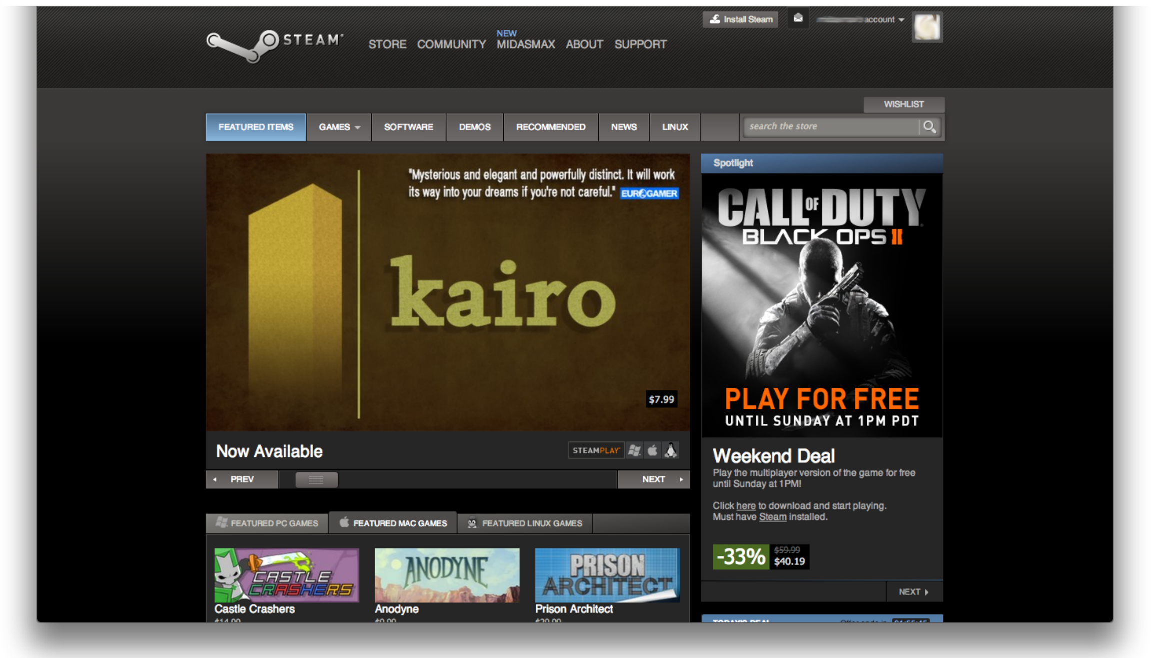
### Virtual Items: Hats

The focus of this thesis is on analyzing one particular category of virtual items in TF2. The items are called **hats**, a head accessory which players may choose to equip on their in-game character. They are one of the most popular virtual items for players in TF2. Obtained through various means (e.g., randomly as players play, as promotional items in tandem with game releases on *Steam*), most hats are limited in supply, which becomes quite apparent to players once promotions end and they cease to be distributed.

The virtual economy of hats was estimated to be worth around $50 million dollars in 2011 [40], prompting *Valve* to hire an economist to manage their in-game economies [59]. Equipping avatars with these hats forms an interesting case of self-expression, since the decision to equip a particular hat for a character class does not improve player attributes within the game to gain an advantage. Instead, diverse hats seem to be equipped based upon issues such as scarcity, style, personal taste, and other subjective factors. This diversity is a result of their method of acquisition/distribution, their customization capabilities (i.e., by color), and their rarity.

### 2.3.2 The *Steam* Platform

*Steam*[4] is an integrated game distribution platform and social networking site (along with some additional functionality). *Steam* allows users to manage their collections of games purchased using it. *Steam* requires users to sign up for a *Steam* account with a unique *Steam Id* in order to create individual *Steam Profiles*. The games distributed on *Steam* include both *Valve*-published and third-party published titles. Fig 2-10 shows a screenshot of the main page of the *Steam* store.



"Mysterious and elegant and powerfully distinct. It will work its way into your dreams if you're not careful."

Play the multiplayer version of the game for free until Sunday at 1PM!

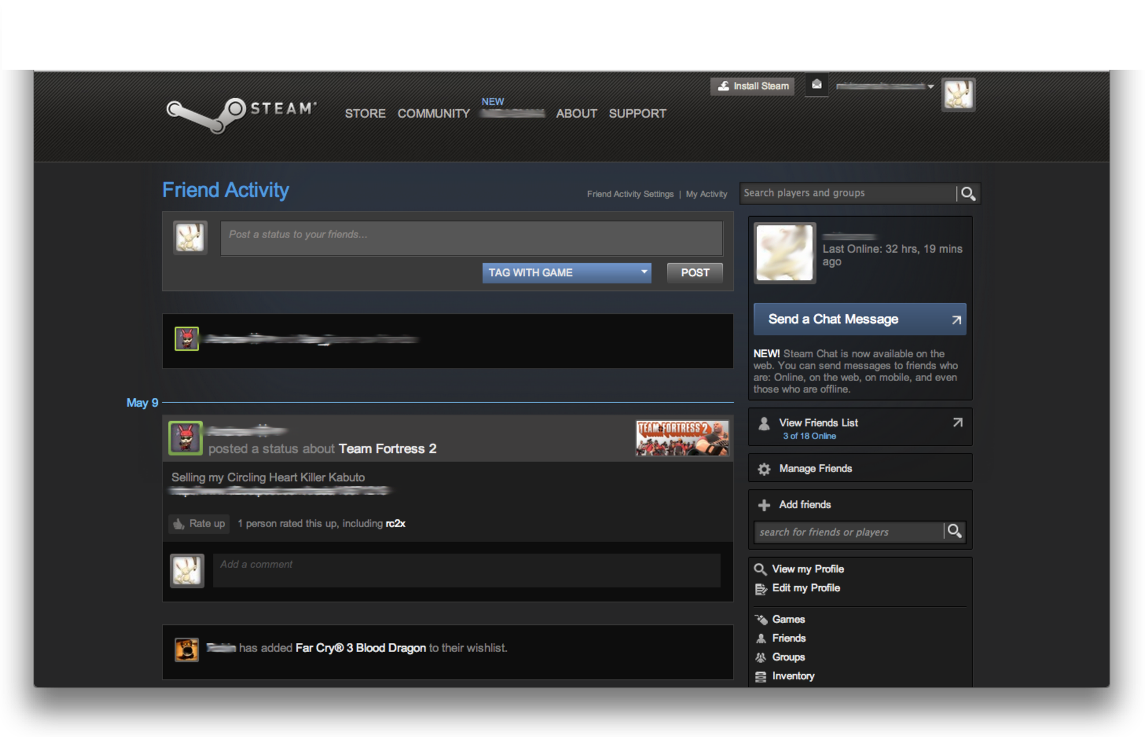
Click here to download and start playing. Must have Steam installed.

Figure 2-10: A screenshot of the *Steam* platform

In terms of social networking, players connect to one another through their "friends lists." Once friends, this enables them to send one another messages, view *Steam* profiles, or find others to play with. Players may also create, manage and join "groups," which are communities of players with similar interests. These social aspects of *Steam* are presented using *Steam Community Pages*—web pages which present familiar social networking capabilities such as a wall for posting messages, and a gallery of

[4] Official Website: `http://store.steampowered.com`

pictures. An example of a *Steam Community Page* for a user is shown in Figure 2-11.



[redacted] Last Online: 32 hrs, 19 mins ago

NEW! Steam Chat is now available on the web. You can send messages to friends who are: Online, on the web, on mobile, and even those who are offline.

[redacted]

[redacted] posted a status about Team Fortress 2

Selling my Circling Heart Killer Kabuto [redacted]

1 person rated this up, including rc2x

[redacted] has added Far Cry® 3 Blood Dragon to their wishlist.

Figure 2-11: A screenshot of a *Steam Community Page* for a user.

*Steam* also allows users to connect to other social networking applications, such as Facebook. In 2011, there were approximately 82.2 million friendship edges[5], 1824 games and 1.98 million groups [2]. At the time of writing, the number of player's concurrently active on Steam is between two and four million. Its network size, along with its gamer-centric demographic, makes it an interesting domain in which to research the relationships between social network behavior and player gameplay.

## 2.4 Areas of Application

In this section, I provide an overview of how the theoretical framework has been applied in our approaches, methods, and results.

[5] The connectivity of users on *Steam* is computationally represented as a graph, with nodes consisting of user profiles or pages, and edges between them representing friendship.

### 2.4.1 Application of Computational Identity Components

In Table 2.2, I summarize how the work in this thesis, pertaining to the implemented systems, relates back to the shared technical underpinnings of computational identity applications identified by Harrell [26].

| AIR Key Concept | Application Area |
|---|---|
| Computational Identity Applications | The two application domains span different types of computational identity applications.<br>The first, *Steam*, is a social networking platform with user identity represented as *Steam* profiles.<br>The second, Team Fortress 2 (TF2) is an online multiplayer videogame, where users engage in a virtual environment with player characters visually represented using 3D models. Avatar customization provides additional avenues for self-expression and identity representation. |
| Computational Identity Components | The system collects and analyzes publicly available data on players' avatar customization with hats, which are represented using 3D models (Modular graphical models).<br>Users on the *Steam* network represents themselves using *Steam* profiles, where each profile consists of textual descriptions (Flat text profiles), and information about their status within the system including, for example, a user's number of friends or number of owned games (Statistical/numerical representation).<br>Both profiles and groups within *Steam* additionally use make use of screenshots, avatar icons, and group profile images (Static media assets). |

Table 2.2: Table summarizing application areas of the theoretical framework relating to computational identity components.

### 2.4.2 Application of the AIR Model

In Table 2.3, I summarize how the work in this thesis, pertaining to the implemented systems, and its results, relate back to the cognitively-grounded AIR model.

| AIR Key Concept | Application Area |
|---|---|
| Cognitive Mechanisms | As opposed to defining categories by hand, or based on existing structures, the approach in this work uses an instance-based learning approach in identifying groups and categories of users and players. These categories are constructed using cluster analysis, using the intrinsic distribution of users to construct categories to classify each user. These clusters are mathematically defined by their means, an abstract "member" of the category, which is used to help determine membership of other individuals. This metaphorically corresponds to prototype effects in cognitive categorization. |
| Social Structures and Interactions | The implemented system analyzes data collected based on both the topological features of a user's social network, which provides insight into the structures comprising one's social identity. Also, it analyzes interactions made by users, including user-to-user communication, and interaction involving the transfer of items and goods. Also, using community derived real-world monetary values for virtual items provides a way to quantify the construction of these values based on these interactions and infrastructures. |
| Computational Extensions | Building upon the data collected and analyzed, the work here also introduces computational extensions to identity representation in the form of **meta-ties**, which are abstract representations of a player's social network which may be both quantitatively and qualitatively analyzed. The second construct that this work introduces is that of player performance through **status performance**, a construction used to model a player's exhibited preference with influence of real-world monetary values. |

Table 2.3: Table summarizing application areas of the theoretical framework relating to the AIR Model

## 2.5 Summary

In this chapter, I have covered the background information motivating the work in this research. Covering the theory, implementation, and related work over the research disciplines of sociology, cognitive science and computer science, this chapter has provided an interdisciplinary basis of understanding for the rest of this thesis, beginning with the theoretical framework. The application domains have been listed and described, and their appropriateness as an interesting and challenging area of application have been highlighted.

Next, I have covered the theoretical framework that underlies the system design and approaches used in the work presented in this thesis. I presented a high-level overview of how the theoretical framework corresponds with the choices and implemented systems. With this high-level understanding of the work, cognitively grounded in the AIR model, the next chapter presents the details regarding the methods used by *Steam-PPA* to calculate and group players based on status performance and model a user's social network structure.

# Chapter 3

# Methods

This chapter covers both the *Steam-PPA* and *AIR-SPC* system. Section 3.1 covers the data-collection that was performed as part of this work, and details the system design of the *Steam-PPA* and *AIR-SPC* systems. Section 3.2 defines **status performance** and discusses how it is derived from monetary values of virtual items in TF2. The section also discusses how the *Steam-PPA* system makes use of its various components in order to calculate the real-world monetary value of virtual items in *Steam* and TF2 along with how *AIR-SPC* uses AI techniques to categorize players according to their status performance.

Section 3.3 goes into detail about how a player's social network in *Steam* is analyzed, extending upon previous work done with tie-strength in social networks. It details how **meta-ties** are created, along with the techniques that were used to construct them, such as the natural language processing capabilities of *AIR-SPC*. Section 3.4 covers with the machine learning techniques that *AIR-SPC* implements. In that section, I also describe the theoretical and algorithmic approaches to perform supervised learning in order to construct a model of the data. This enables the system to perform prediction and categorization using support vector machines (SVM).

## 3.1 Data Collection

In this section, I describe the *Steam-PPA* system in detail. First, starting with the overall system design, I highlight its capabilities for advanced data collection of public user profile information on *Steam*. The robustness of the system arises from its capabilities to collect public information not just from the *Steam API*, but also from the *Steam Community Web Pages* and other third-party websites, with a single, common interface. Some of the its data-processing capabilities are relatively complex and the amount of data required for profile analysis exponentially increases with the number of profiles to be analyzed. For example, a factor such as the number of mutual friends that a user has requires tens of thousands of player profiles to be queried in order to make the comparisons.

### 3.1.1 System Design

The *Steam-PPA*'s system is composed of three main layers, the **network** layer, **caching** layer, and the **computation layer**. Figure 3-1 depicts an overview of the system, which is constituted by three main components.

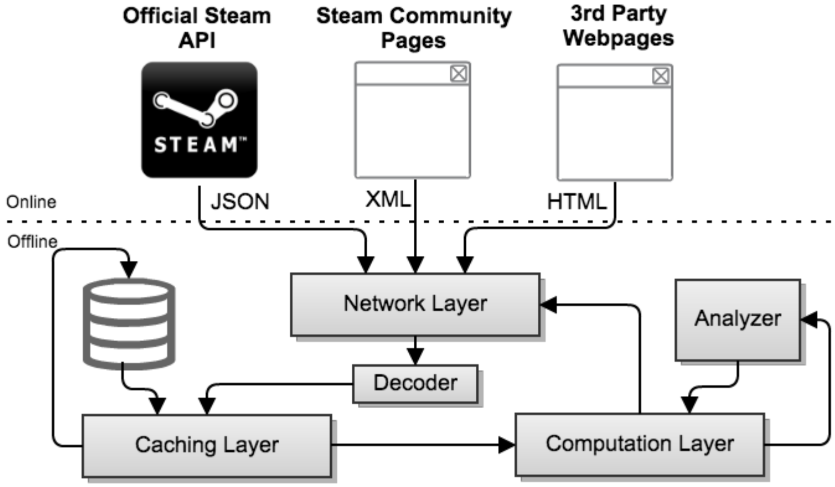


Figure 3-1: An overview diagram of the implemented system, made up of three main layers (Network, Computation and Caching layers). Data is collected from various sources through API calls or web-scraping using HTTP requests. Different analyzers may be implemented to make use of the data.

### Network Layer

The network layer makes requests to remote servers online in querying for game, social network or player information. The decoder is used to parse and extract appropriate information from the results of the queries. There are three main types of remote servers:

1. **Steam Web API**: The API is an official and free service provided by *Valve* that grants access to information about games on the Steam network (e.g., schema of all the items in TF2) and player profiles on the steam network (e.g., summary of player avatar name, player's friends). Registration is required to use the API and receive a *Steam Web API* key, which is required to make requests. Requests are limited to 100,000 API calls per day, and are returned in JSON format.

2. **Steam Community Pages**: These HTML pages are maintained and owned by *Valve*. They provide common social networking capabilities (e.g., wall posting, picture uploading, user-to-user messaging); and some portions are retrievable in XML format, while others have to be manually scraped as HTML pages.

3. **3rd-Party Webpages**: These refer to any external sites that are unaffiliated with *Valve* or *Steam*, but may contain publicly aggregated information about TF2 players. One site that was used as part of system is **Backpack.tf**[1], a site that crowd-sources prices on all the items available in TF2. They have to be manually scraped as HTML pages.

### Caching Layer

To keep within the Steam API limits and prevent unnecessary queries, I implemented a caching layer that stores all the decoded data received, via the decoder from the network layer, onto the hard disk. Timestamps are added to the cached data and stored as either JSON or XML files. The caching layer is also used to store intermediate results from the computation layer, such as the computed hash table containing

[1] Backpack.tf: `http://backpack.tf`

prices for each item, for efficiency.

#### Computation Layer

The computation layer interfaces with both the network and caching layers and is responsible for computing results from the data, which are the caching layer. For instance, in determining the number of common applications that a user shares with each of the users friends, it needs to get the friend list of the player (Steam API) and the lists of each players and their friends application list (from the Steam Community Page). Section 3.2.1 outlines the more complex scenario of calculating the real-world monetary value of a players customized avatar using a combination of data sources, as well as how the computation layer calculates various sets of values based on metrics constituting tie strength.

## 3.2 Status Performance

In this section, I cover how both the *Steam-PPA* and *AIR-SPC* are used in order to construct a quantitative notion of players' preference in customizing their avatars in TF2. The chosen area of focus, as described in Section 2.3.1, is on avatar customization preferences exhibited by players with **hats**. I associate each hat with their real-world monetary values by using *Steam-PPA* to data mine publicly available information from a 3$^{\text{rd}}$-party website, which provides up to daily updated and crowd-sourced real-world monetary values for each hat. By assigning real-world monetary values to items owned by players, they construct a form of performance related to value, which I term as **status performance**.

### 3.2.1 Status Performance using Avatar Hat Customization

The first step was to provide a computational structure to quantify how a player "performs" self-expression in TF2 by customizing his or her character using hats. Each hat has a calculated real-world monetary value, and I distinguished between two ways of possessing them. First, the **inventory** hat value refers to the total monetary value of hats in a players inventory (or backpack, using TF2 terminology). Secondly, the **equipped** hat value refers to the total monetary value of hats actively equipped across all of the players characters, across the 9 character classes. With these two values, I defined the **status performance** value as a tuple of the form: `(equipped-value, inventory-value)`.

Calculating the monetary value of a hat requires *Steam-PPA* to query for the list of all items a player possesses via the *Steam API*. It then performs a filtering for only hats by checking the `type` attribute of the results. Then, it extracts data from a third-party price-listing website to get the prices for both regular and unusual hats using a combination of HTML parsing, price calculations and look-ups. Regular hats require a single look up (item `defindex`), while unusual hats require two look-ups (item `defindex` and `particle-id`), in order to determine their values. Figure 3-2 shows a flowchart of *Steam-PPA* performing the necessary queries in order to calculate the value of any hat a player possesses.

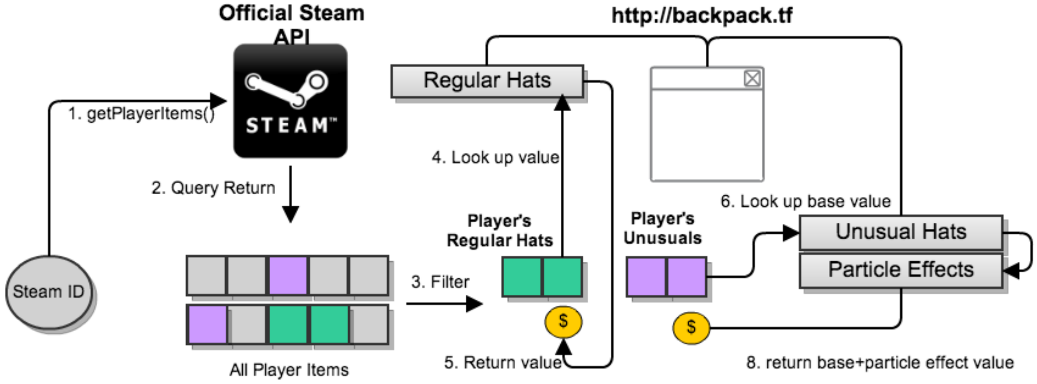


Figure 3-2: Flowchart illustrating how Steam-PPA performs several look-ups in order to calculate the monetary value of different types of hats. (caching not shown)

### 3.2.2 Categorizing Players using k-means Clustering

To ameliorate the variation in calculated status performance across the dataset of players, I performed the **k-means algorithm** [39] for cluster analysis in order to categorize the players into distinct and separate groups. Each cluster (or group) $k$, is defined by a central “abstract” member, methematically represented as the mean value $m^{(k)}$. Next, I used an iterative refinement technique as follows:

1. **Assignment Step**: Each data point $x^{(i)}$ is assigned to the cluster $k^i$ with the closest mean. Mathematically, this is represented as:

$$\hat{k}^{(i)} = \arg\min_k \left\{ distance(m^{(k)}, x^{(i)} \right\}$$

2. **Update Step**: Each cluster's mean is updated to be the centroid of all the data points within it. Mathematically, this is represented as:

$$m^{(k)} = \frac{\sum_{x_i \in \hat{k}^{(i)}}}{|\hat{k}^{(i)}|}$$

The two steps are repeated until the assignments of the data points no longer change or some defined threshold is reached.

In applying this to the status performance value tuples, the distance measure is the **euclidean distance** between each datapoint. Letting $x$ and $y$ represent the status performance data points of two separate players, I mathematically represented the distance between status performance datapoints using the equation:

$$distance(x, y) = \sqrt{(equipped_x - equipped_y)^2 + (inventory_x - inventory_y)^2}$$

This approach ultimately enables the categorization of the data points according to the cluster each belongs to.

### Determining the Number of Clusters

For the k-means algorithm, the key parameter is the number of clusters $k$. Identifying the best value of $k$ requires the balancing of two trade-offs: 1) being able to minimize the distance between each data point $x^{(i)}$ and its associated cluster mean $m^{(k)}$ and 2) not over-fitting the data points to prevent generalization in the event adding new data points. This trade-off requires both a quantitative way to calculate the total separation distance of each point to its clusters mean, but also a qualitative/intuitive reasoning to set a threshold to the number of clusters, since, simply specifying $k$ to be the number of datapoints would give total separation of each point to its mean as 0.

Thus, I made use of the within-group sum-of-squares (WSS) in order to determine the total separation distance of each point to its cluster mean. The formula for the within-sum-of-squares is defined using the formula:

$$WSS_k = \sum_{x_i \in \hat{k}^{(i)}} (x^{(i)} - m^{(k)})^2$$

Next, I varied the number of specified clusters between a range, which would give a mixture of WSS values per specified cluster size. The idea is to then analyze the variation of WSS values and then determine the best (though not necessarily optimal) number of clusters to describe the dataset of status performance values. Algorithm 1 describes the process of aggregating the WSS values per tried cluster.

**Algorithm 1** Algorithm for cluster determination using within-group sum of squares.

```
for $k = 2 \to 15$ do
    $means \leftarrow kmeans(values, k)$
    for $c = 0 \to |means|$ do
        $Errors[k] \leftarrow WSS(values_c, means[c])$
    end for
end for
```

Additionally, I made use of the Bayesian Information Criterion (BIC) as a second test to decide on the best number of clusters. This entire process allows for the mapping of each data status performance value onto a smaller set of categories, each defined by a cluster. Ultimately, it presents an easier to describe the distribution of status performance of players across the dataset in terms of discrete labels.

## 3.3 Meta-ties

In this section, I describe the process in which the *Steam-PPA* system collects data about players' social networks using quantitative factors describing the relationship that players have with one another, called **tie strength**. First, I provide details on the twelve factors extracted from players on the *Steam* network in order to estimate tie strength values. Second, I describe the process of decomposing the dataset of features (250×12) into a smaller, more succinct version (250×5), using **principal component analysis** in a process called dimensionality reduction. Third, I introduce the notion of **meta-ties**, which describe the resulting five factors (principal components) and describe them as a way of reasoning about the dataset in a abstract, domain-specific, and humanly-interpretable way.

### 3.3.1 Tie Strength in the *Steam* Network

Gilbert, et al. showed that tie strength between users in social networks could be predicted by using a combination of **predictive variables** (e.g., users' numbers of friends, numbers of words exchanged, etc.) [19]. These predictive variables are grouped into 6 different tie strength **dimensions**. Granovetter identified the first 4 dimensions: *intimacy*, *intensity*, *duration* and *reciprocal services* [23]. Next, Ronald Burt extended the list with *structural* variables, which covers network topology [6]. Lastly, Barry Wellman and Scot Wortley introduced *emotional support* variables, used primarily for defining strong ties [60]. The *Steam-PPA* system's implementation differs from Gilbert and Karahalios by not making use of the *social distance* predictive variable, identified by Nan Lin, et al. It covers aspects such as socioeconomic status, education level, and gender [36], which is information that is neither collected nor provided in *Steam* or TF2.

I implemented the collection of data on the *Steam* network for calculating a total of twelve predictive variables for the six dimensions outlined above. As far as I know, this is a novel application domain of such tie strength measures. Table 3.1 provides the list of the twelve predictive variables.

| # | Dimensions | Predictive Variable |
|---|---|---|
| 1 | Intensity | Own Wall Posts |
| 2 | Intensity | Friend Wall Posts |
| 3 | Intensity | Words Exchanged |
| 4 | Intimacy | Friend Count |
| 5 | Intimacy | $2^{\text{nd}}$ Degree Friends |
| 6 | Duration | Days as Friends |
| 7 | Reciprocal Services | Traded Item Count |
| 8 | Reciprocal Services | Common Applications |
| 9 | Emotional Support | Positive Words |
| 10 | Emotional Support | Negative Words |
| 11 | Structural | Mutual Friends |
| 12 | Strcutural | Common Groups |

Table 3.1: The table above shows the Tie Strength Predictive Variables used in *Steam-PPA* for analyzing the *Steam* Network, along with their dimensions.

Next, I shall provide a detailed description of each of these predictive variables as they pertain to the domains of *Steam* and TF2, as well as of how *AIR-PPA* collects the information. The predictive variables are presented the same order as they are listed in Table 3.1, and I refer each predictive variable back to its index in this table using the "#" symbol within parentheses.

### Intensity Variables

Each player with a public profile possesses a *Steam Community Page*, which has various common social networking features, such as a "wall" for posting messages, and photo albums for uploading and displaying pictures. *Own Wall Posts* (#1) refers to the number of posts that a player adds to his or her own wall, much akin to a status update. *Friend Wall Posts* (#2) refers to wall posts that were added by a player's friend; I calculate the average number. The *Words Exchanged* (#3) variable refers to the total number of words that are on a user's wall.

The *Steam-PPA* system collects the required data for these variables by parsing the *Steam Community Page* of each user with a HTML parser, which isolates tags on the page corresponding to wall posts. It differentiates between posts made by the player or by the players friends by cross-referencing each post's `author-id` against the *Steam ID* of players. It then uses the Python *Natural Language Toolkit (NLTK)*[2] library to perform sentence and word segmentation, which splits the text in each wall post first into sentences, and then into individual words.

### Intimacy Variables

Each player on *Steam* has a list of friends with whom they may communicate with using the system. The first predictive variable used for *intimacy* is *Friend Count* (#4), which is the number of friends which the player has in his list. The second predictive variable is *2nd Degree Friends* (#5), which is the number average number of friends that the player's 1st degree friends have. *Steam-PPA* uses the *Steam API* to query for a player's "friendslist" for calculating these values.

### Duration Variables

A player on *Steam* may become a "friend" with another player by sending a friend request. Once the other player accepts the request, both players are then deemed friends by *Steam.* A user may also choose to decline, ignore, or block a friend request. The *Steam API* sets a Unix timestamp of when this friendship is formed[3], and *Steam-PPA* makes use of this to calculate, for each of the player's friends, how long they have had this friendship (in days). The predictive variable, *Days as Friends* (#6), is the average number of days that a player has been friends with each player on his friend list.

### Reciprocal Services Variables

Unlike other social networking platforms, such as *Facebook* and *Twitter*, there is a good basis to analyze actions performed by users involving the exchange of information, services, or economic goods. In *Steam* and TF2, the trading of virtual items, particularly hats, provides a relevant factor for this dimension of predictive variables.

[2] NLTK 2.0: http://nltk.org/

[3] There exists a bug in the API makes this variable either unavailable, or partially available to either party, but not both. *Steam-PPA* handles this by checking for the variable on both profiles, and taking the larger of the two, if available.

The first predictive variable, *Traded Items Count* (#7), is the number of items in a player's entire list of items which were obtained through trading. This is performed with *Steam-PPA* by first querying for a players entire set of virtual items, and filtering those items which had its `origin` set to `trading`.

Since *Steam* contains the library of applications and games that a user owns, it can be used to calculate the number of applications in common that the user has with each of his or her friends. This results in the second predictive variable in this dimension is *Common Applications* (#8), which is the average number of common applications that a user has. *Steam-PPA* obtains this list by parsing a user's *Steam Community Page* profile's "Games" tab[4]

### Emotional Support Variables

From the wall posts gathered as part of *Steam-PPA*'s analysis of *intensity* variables, the natural language processing component of *AIR-SPC* was used to perform sentence and word segmentation on each wall post in extract individual words. *AIR-SPC* performs sentiment analysis using the `sentiment_classifier` [5] package to analyze each word using word-sense disambiguation from `wordnet` and occurrence statistics from the `movie_review` corpus from NLTK, individual words were classified according to the emotions conveyed, which may be either positive or negative, corresponding the predictive variables *Positive Words* (#9) and *Negative Words* (#10).

### Structural Variables

In analyzing the network structure of a user's network, *Steam-PPA* calculates the number of mutual friends a player has with each of his 1st degree friends by comparing the *Steam IDs* across both list of friends. The predictive variable *Mutual Friends* (#11) is calculated by averaging the number of mutual friends. This measure forms a kind of implicit structural factor of a player's social network.

[4] The *Steam API* was later updated to enable direct queries for a user's list of applications, which *Steam-PPA* was updated to use. The values obtained were similar, and so for consistency, it was reverted back to the parsed version.

[5] Python Package - sentiment_classifier v0.5: `https://pypi.python.org/pypi/sentiment_classifier`

The second network structure analyzed involves “groups” in *Steam*. These groups are user-created and provide a way for users with similar interests to form a community on *Steam*. These groups are publicly displayed on a player’s *Steam Community Page* profile and each group identifies itself with several computational representations, such as a group ID (assigned by *Steam*), display picture, text descriptions, and even self-imposed rules for membership. Using the *Steam API*, *Steam-PPA* may query for a user’s list of groups. It then calculates the number of common groups that a player has with each of his friends. The predictive variable *Common Groups* is calculated by averaging the number of common groups that a player has with his friends.

### 3.3.2 Tie Strength Dimensionality Reduction using PCA

I performed dimensionality reduction on the set of features using Principal Component Analysis (PCA). Given a dataset, performing PCA decomposes the data into several components, each defined as a linear combination of the original set of features using coefficients. Performing PCA on our dataset allows us to 1) reduce the number of features required to describe the dataset, and 2) infer relationships between the original features through the coefficients used in each principal component. Studying the resultant principal components and their coefficients allows for reasoning about the data with more abstract, higher-level terms defined here as the meta-ties of the player. This allows for describing each players social network in meta-ties terms, instead of the original, fine-grained individual tie strength predictive variables.

One may calculate its **scoring** (or score) by using a linear combination of the coefficients obtained from PCA, and multiplying them with the respective predictive variables that they correspond to. This scoring represents a numeric value of the meta-tie, and may be positive or negative in sign($+/-$). Mathematically, the scoring of each meta-tie $j$ is defined as $score_j$, using the equation:

$$score_j = \sum_{i=0}^{12} coefficient_i{}^j \times value_i$$

This value allows us to quantitatively describe the feature set of meta-ties as a feature vectors of numerical values.

## 3.4 Training & Classification

In this section, I first describe the process of determining the presence of a relationship between a players' meta-ties and their status performance. Next, I describe the process of constructing a model of a player using their meta-ties scores. The model is trained to be able to perform prediction of a player's status performance using their meta-ties.

### 3.4.1 Classification using Cluster Association

The first step I took in studying the relationship between meta-ties and status performance was to analyze the degree of association of the discrete status performance labels with meta-ties. I performed k-means clustering on the resulting meta-ties from PCA and then used coefficients of associations tests to study any relationship between both set of clusters.

### 3.4.2 Classification using Support Vector Machines

Support Vector Machines (SVMs) are a form of supervised learning methods that can be used for classification or regression problems. In a binary classification example, I would train the SVM on a labeled dataset and, if they are linearly separable, the SVM will find a unique separation boundary in the form of a hyperplane with points falling on each side having different classifications. The separation boundary would be one in which the margin is maximized.

In general, not all data points will be linearly separable often as a result of overlapping class-conditional probabilities. Also, there is a chance of over-fitting on the training data points, which might negatively affect the generalizability of the classifier for future points. As such, I use *slack variables*, which results in a 'soft margin' allowing some data points to be incorrectly misclassified with a certain penalty with the aim of overcome over-fitting. The general formulation of SVMs as constrained quadratic programming problem is as follows:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & C\sum_{i=0}^{n}\xi_i + \frac{1}{2} \parallel \theta \parallel^2 \\ \text{subject to} \quad & y_i(\theta \cdot \mathbf{x}_n + \theta_0) \geq 1 - \xi_i,\ i = 1, \ldots, m \end{aligned}$$

where $x_i$ represents each training data point, with $y_i$ being its corresponding target classification. $\theta$ is the model, or parameter, of the classifier with offset $\theta_0$, while C and $\xi$ represent the penalty and slack variables respectively.

### Multi-classification for Status Performance Labels

The approach taken makes use of SVMs to classify players using their meta-ties into several discrete status performance labels. Since SVMs are actually binary classifiers, in order to perform multi-class classification, the approach is to train multiple binary classifiers, one for each individual target status performance label. Given an input vector of meta-ties, each trained classifier gives its predicted status performance label and the next step is to employ **voting** as a way of deciding the best classification result for the data. The common voting strategies to decide on a classification described as follows:

- In **one-versus-one** voting, a single SVM is trained for *every possible pair* of target status performance labels. Thus, given $k$ possible status performance labels, it would require $k^2$ SVMs. Given an input vector of meta-ties to be classified, selection is done by picking the mode: the status performance label which occurs the most.

- In **one-versus-rest** voting, a single SVM is trained for *every* target status performance label. Thus, given $k$ possible status performance labels, the system will only require $k$ SVMs, one for each status performance label. When making the prediction, the class with the highest classification output is chosen. This is defined as a *winner-takes-all* strategy.

The approach used in *Steam-SPC* is the latter, in which **one-versus-rest** is used, as it is quicker to construct the model, and is sufficient enough.

## 3.5 Summary

In this chapter, I have outlined the various methods, algorithms, and experimental designs that were employed in both the *Steam-PPA* and *AIR-SPC* systems. The combination of algorithmic approaches and implemented systems have allowed the collection, computation, and analysis of the required data in the target application domain. The next chapter presents the results from these systems and experiments.

# Chapter 4

# Results

In this chapter, I present the results from the methods and experiments that were presented in Chapter 3. Most of the results are presented in summarized, tabular, or graphical form, in order to best convey them for reasoning and analysis (which I cover in Chapter 5). In Section 4.1, the results from the status performance data collection from *Steam-PPA* are presented, including the distribution of hat monetary values across the dataset and results of the clustering analysis in order to construct the status performance categories and labels. Section 4.2 covers the results of the data collection of tie strength predictive variables from the dataset, along with results of performing dimensionality reduction using PCA, and the quantitative representation of meta-ties obtained using the *AIR-SPC* system. Section 4.3 presents results from the construction of a model of the data by *AIR-SPC*, through instance-based learning over the dataset of players, each represented as a feature vector of meta-ties, and the prediction performance of both the k-means algorithm and support vector machines (SVM).

## 4.1 Status Performance

In this section I first present the real-world monetary value of the hats owned by each player in our dataset. Next is the distribution of the data for both the equipped and inventory monetary values. Next, the construction of the status performance tuples of `(equipped, inventory)`data points are summarized. Finally, I present the results of the clustering analysis on these data points. Each resultant cluster is defined in terms of a central data point. The proximity of the player data points to this central data point allows the assignment of status performance labels to each player.

### 4.1.1 Monetary Value of Hats

Figure 4-1 shows a scatterplot of the (logarithmic) inventory value versus equipped value of the virtual hats across the player profiles. Table 4.1 gives a numerical summary of the distribution of inventory and equipped virtual hat values.

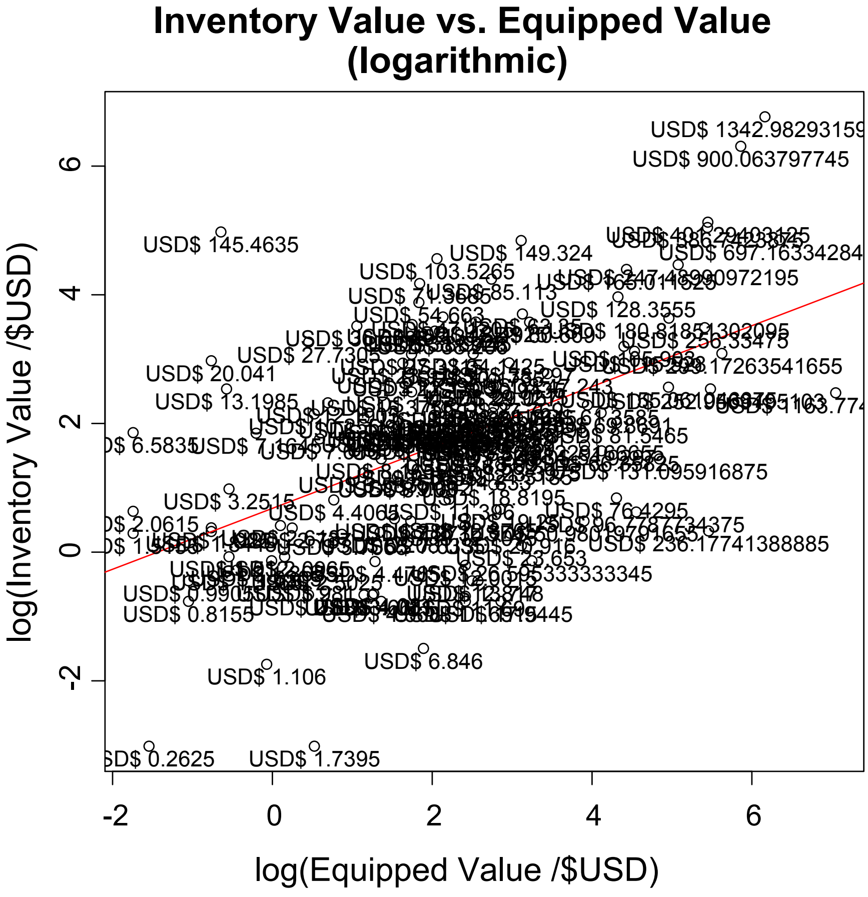


Figure 4-1: Scatter plot showing the comparison between the log-values of inventory items versus the log-values of equipped items for each player based on their profiles. Each data point's value is the sum of the inventory and equipped values of each player.

|  | Mean | Median | Std. Dev | Min | Max |
|---|---|---|---|---|---|
| Equipped | \$25.83 | \$1.52 | \$119.89 | \$0.00 | \$1151.88 |
| Inventory | \$15.79 | \$1.46 | \$84.82 | \$0.00 | \$867.87 |

Table 4.1: Summary of Predictive Variables across the Dataset.

### 4.1.2 Clustering Players by Status Performance

For each player, status performance is represented using a computational data structure, which is a 2D data point of the form: `(equipped, inventory)`. Using the k-means algorithm, I performed clustering on the dataset to obtain a small number of discrete, nominal clusters of which each data point belongs to. In determining the number of clusters for the k-means algorithm, the number of clusters was varied between one and fifteen, and used the within-group sum of squares as a measure of an ideal number of clusters. The variation of the within-group sum of squares across the number of clusters is plotted in Fig 4-2. Additionally, I also used model selection according to the Bayesian Information Criterion (BIC) for Expectation-Maximization (EM) to determine the ideal number of clusters.

From the results, I chose $k = 4$ to be the ideal number of clusters (BIC= -1997). Even though EM model selection returned k=8 (BIC= -1925) as the best value, overfitting appears to occur after $k = 7$ clusters (Figure 4-2). Setting $k = 4$ has similar BIC values as when the k-value was varied between four to seven. Next, I hand-assigned labels to each cluster, defining the **status performance** labels, the results of which are shown in Table 4.2. Figure 4-3 shows the clusters in the projected inventory-value/equipped-value graph, where each `(equipped, inventory)`data-point is classified according to the nearest cluster mean.

|  | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
|---|---|---|---|---|
| Equipped | \$0.09 | \$6.91 | \$36.82 | \$301.06 |
| Inventory | \$0.33 | \$4.42 | \$27.12 | \$167.89 |
| Frequency | 91 | 107 | 13 | 39 |
| Status Performance | `NONE` | `LOW` | `MEDIUM` | `HIGH` |

Table 4.2: The table shows the assignment of labels for each status performance cluster, based on the values of the mean data point.

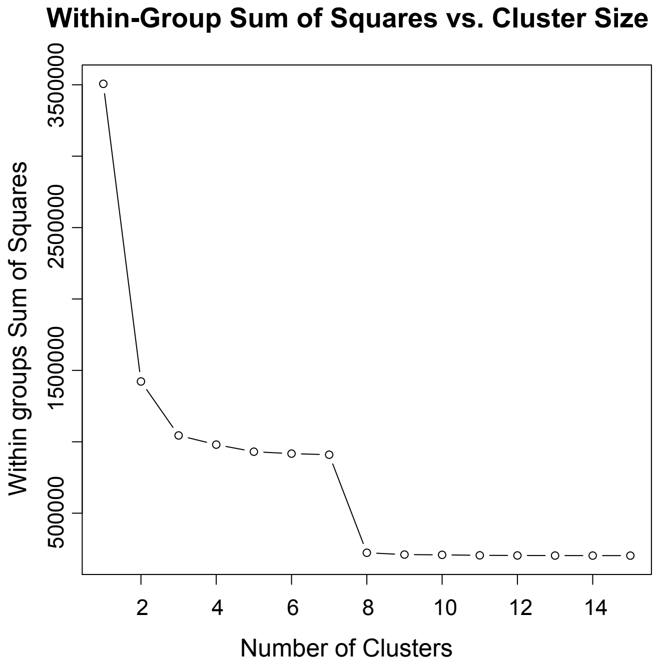


Figure 4-2: The graph shows the within-group sum of squares error plotted against the number of status performance clusters.

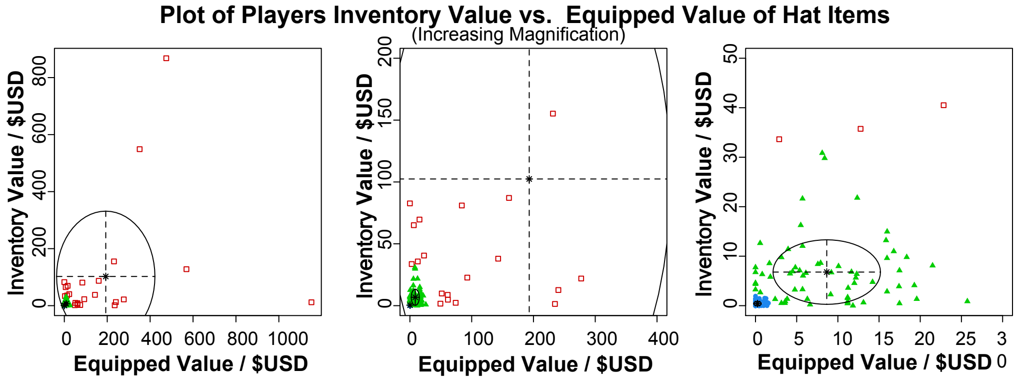


Figure 4-3: Cluster classification using the the 3 clusters obtained, with increasing magnification to the right for visibility.

## 4.2 Meta-ties

In this section, I present the results from *Steam-PPA*'s data collection of tie strength predictive variables for the two-hundred and fifty player profiles on *Steam*. Next, I present the results of the dimensionality reduction performed by *AIR-SPC*, and show how the resulting principal components and their representations are used to construct meta-ties.

### 4.2.1 Predictive Variables

Table 4.3 contains a summary of the distribution of each of the twelve tie strength predictive variables that were calculated using *Steam-PPA*. The results presented are not normalized in order to provide a clearer understanding of the scale and magnitude of each predictive variable and its distribution across the dataset.

| Dimension | Predictive Variable | Mean | Med. | Std. Dev | Min | Max |
|---|---|---|---|---|---|---|
| Intensity | Own Wall Posts | 0.896 | 0.00 | 2.30 | 0 | 19 |
| Intensity | Friend Wall Posts | 8.456 | 1.00 | 12.92 | 0 | 50 |
| Intensity | Words Exchanged | 79.37 | 11.00 | 121.73 | 0 | 600 |
| Intimacy | Friend Count | 92.63 | 70.50 | 75.87 | 1 | 299 |
| Intimacy | 2$^{\text{nd}}$ Degree Friends | 70.22 | 65.74 | 30.37 | 1 | 144.62 |
| Duration | Days as Friends | 690.70 | 670.89 | 357.70 | 1 | 1641 |
| Reciprocal Services | Traded Item Count | 9.65 | 0.00 | 29.68 | 0 | 271 |
| Reciprocal Services | Common Apps. | 13.85 | 10.38 | 12.65 | 1 | 95.97 |
| Emotional Support | Positive Words | 76.26 | 11.00 | 113.58 | 0 | 580 |
| Emotional Support | Negative Words | 3.10 | 0.00 | 6.52 | 0 | 34 |
| Structural | Mutual Friends | 5.94 | 3.98 | 5.89 | 0 | 32.24 |
| Strcutural | Common Groups | 1.02 | 0.73 | 0.97 | 0 | 4.98 |

Table 4.3: Predictive Variable Summary of Collected Profiles.

### 4.2.2 Principal Components of Predictive Variables

With each player profile's set of predictive variables calculated and analyzed using *Steam-PPA*, they are then normalized in preparation for dimensionality reduction using PCA. Table 4.4 shows the top five scoring principal components obtained from the PCA, along with their individual standard deviation (1$^{\text{st}}$ row) and variance of the distribution (2$^{\text{nd}}$ row). The reason for choosing the top five is to ensure that they cumulatively cover at least 80% of the variance of the distribution (3rd row). As the table shows, the top five cover 85% of the variance of our dataset. Visually, the principal components can be ranked using by plotting a scree-plot and this is shown in Figure 4-4.

| | MT #1 | MT #2 | MT #3 | MT #4 | MT #5 |
|---|---|---|---|---|---|
| Standard Dev. | 2.3833 | 1.2272 | 0.9845 | 0.8943 | 0.8942 |
| Prop. of Variance | 0.4734 | 0.1255 | 0.1078 | 0.0808 | 0.0666 |
| Cumulative Prop. | 0.4734 | 0.5989 | 0.7067 | 0.7875 | 0.8541 |

Table 4.4: The Top Meta-Ties and their Data Coverage.

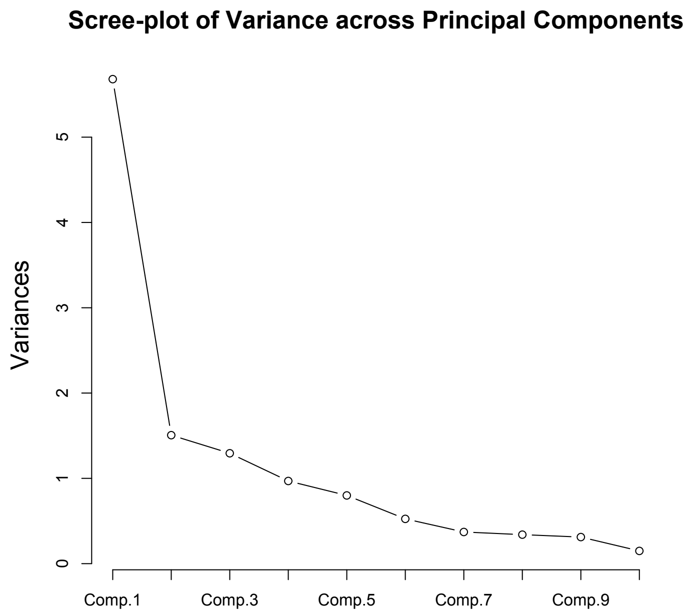


Figure 4-4: Scree-plot showing the how much each of the various meta-ties (principal components) cover the originl data in terms of variance.

### 4.2.3 Meta-tie Coefficients and Scores

Each principal component is derived based on the normalized values of the predictive variables from the dataset. From this point onward, I shall refer to them as **meta-ties**. Using the method described in Section 3.3.2 of the Methods chapter, the $score_j$ is calculated using the summation over all 12 predictive variables, where each predictive variable $i$ has its value $value_i$ multiplied against the the corresponding coefficient for meta-tie $j$, $coefficient_i{}^j$. Table 4.5 lists the coefficients for each predictive variable, for each meta-tie. Cells which are empty indicate that the predictive variable does not form any part of the meta-tie's composition.

| Dimension | Predictive Variable | MT #1 | MT #2 | MT #3 | MT #4 | MT #5 |
|---|---|---|---|---|---|---|
| Intensity | Own Wall Posts | -0.269 | -0.332 | 0.401 | -0.350 | |
| Intensity | Friend Wall Posts | -0.375 | -0.123 | -0.177 | 0.198 | |
| Intensity | Words Exchanged | -0.389 | -0.256 | -0.186 | | |
| Intimacy | Friend Count | -0.294 | 0.302 | 0.298 | -0.278 | 0.230 |
| Intimacy | 2$^{nd}$ Degree Friends | -0.288 | 0.284 | -0.352 | | |
| Duration | Days as Friends | 0.136 | | 0.505 | -0.574 | 0.441 |
| Reciprocal Services | Traded Item Count | -0.163 | 0.334 | 0.275 | 0.510 | 0.555 |
| Reciprocal Services | Common Apps. | | | 0.713 | 0.233 | -0.531 |
| Emotional Support | Positive Words | -0.389 | -0.252 | | | |
| Emotional Support | Negative Words | -0.353 | -0.325 | 0.132 | | |
| Structural | Mutual Friends | -0.245 | 0.427 | | -0.300 | -0.346 |
| Structural | Common Groups | -0.292 | 0.407 | | | -0.181 |

Table 4.5: Table showing coefficients defining each Meta-Tie.

A thorough analysis of each meta-tie and the reasoning behind the coefficient values and what they represent in terms of the player population is covered in Section 5.2 of the Analysis chapter.

## 4.3 Predicting Status Performance using Meta-ties

In this section, I present the results from *AIR-SPC*'s approach in constructing a model of the distribution of players. Constructing a model involves using the meta-ties from Section 4.2 as features, and subsequently using the model for prediction of the status performance labels, from Section 4.1, for each player.

First, I present results of performing k-means clustering over the players using the 5-dimensional (5D) data points, with each dimension corresponding to the scoring of the player for each meta-tie. Next, I present the results for tests used to study the association between the resulting meta-ties clusters and status performance clusters. Finally, I present the results of using support vector machines to train a model through instance-based learning. The results contain the set of optimal parameters used for the model, as well as the classification performance of the model.

### 4.3.1 Clustering Players by Meta-ties

Similar to the clustering approach using k-means over status performance in Section 4.1, I performed k-means clustering over the dataset of meta-ties to categorize players according the meta-ties. Similarly, I determined the ideal number of clusters by using the BIC score as our model selection measure, along with the within-group sum of squares, by varying the number of clusters to be between two and fifteen. I obtained an optimal value of $k = 4$ clusters (BIC=-2986), and the within-group sum of squares error plotted against the number of clusters is shown in Figure 4-5.

### 4.3.2 Associating Meta-ties and Status Performance

With clusters of players for both meta-ties and status performance, the results of visually representing the correspondence between both types of clusters is shown in Figure 4-6. It shows a how the meta-tie clusters are distributed over the dataset by projection over meta-tie #1 and meta-tie #2. Next, each data point on the graph is identified with the status performance label that it was assigned from status performance clustering process. A quantitative representation of this is shown as a cross-tabulation in Table 4.6, which describes the overlap between the meta-tie clusters and status performance clusters.

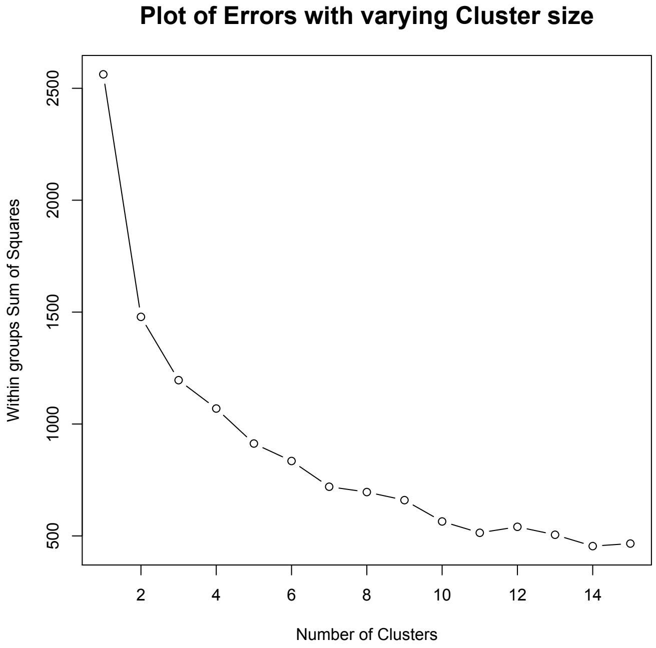


Figure 4-5: The graph shows the within-group sum of squares error plotted against the number of clusters in determining the ideal number of meta-ties clusters

| | NONE | LOW | MEDIUM | HIGH |
|---|---|---|---|---|
| Meta-ties Cluster 1 | 31 | 34 | 0 | 5 |
| Meta-ties Cluster 2 | 49 | 11 | 0 | 3 |
| Meta-ties Cluster 3 | 9 | 42 | 2 | 13 |
| Meta-ties Cluster 4 | 2 | 20 | 11 | 18 |

Table 4.6: Crosstab of Meta-ties Principal Components against Status Performance Labels

### 4.3.3 Classification using Support Vector Machines

The next step was to perform learning using *AIR-SPC* over the feature dataset of meta-ties and target classification values of status performance labels. Due to the size of dataset, cross-validation was employed in order to not divide the data into equal and separate training and test sets, which would reduce the amount of training data available by half. Obtaining $k = 4$ as the number of clusters for both meta-ties and status performance, I employed 4-fold cross-validation, and specifically stratified 4-fold cross-validation due to the uneven number of data points across the status performance clusters(13 for cluster 3 and 107 for cluster 2).

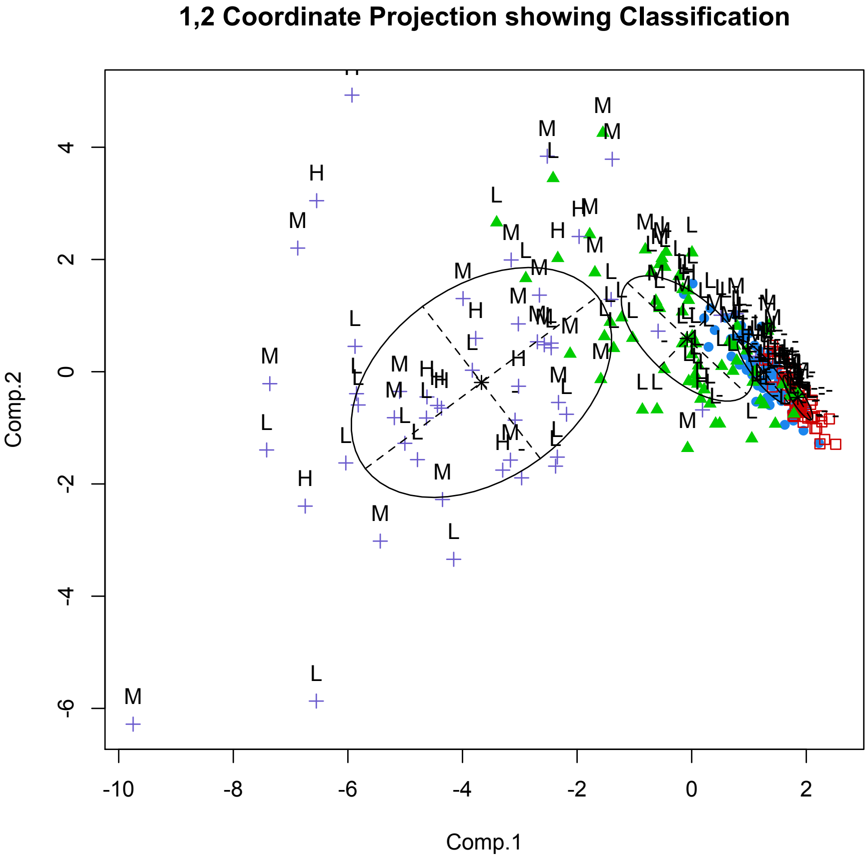


Figure 4-6: Scatterplot of the players when plotted in principal component dimensions corresponding to the top two principal components. Five clusters determined by cluster analysis are shown as ellipses, with the color and symbol of the point corresponding to which cluster each point belongs to. The numerical label identifies the status performance label for each player.

In order to obtain the optimal values for our SVM parameters, I performed a grid-search with 4-fold cross-validation, and varied the parameters of the SVMs as follows. For the kernels, I experimented with the linear kernel and the radial basis function kernel. I varied the C-value in the range [0.01, 0.1, 1, 10, 100, 1000] for both kernels, and changed the tolerance $\gamma$ in the radial basis kernel in the range of [0.001 and 0.0001]. The grid-search was performed to optimize the paramters across three measures of predictive accuracy, namely the prediction, recall and F-1 score.

The best performing parameters are listed in Table 4.7. The optimal parameters for the SVMs were the `rbf` kernel, $C = 100$ and $\gamma = 0.1$. The performance of the model was validated using stratified 4-fold cross-validation, and achieved an accuracy of 58%.

| | Precision | Recall | F-1 Score | #Samples |
|---|---|---|---|---|
| NONE | 0.74 | 0.89 | 0.81 | 91 |
| LOW | 0.84 | 0.79 | 0.81 | 107 |
| MEDIUM | 1.00 | 1.00 | 1.00 | 13 |
| HIGH | 1.00 | 0.72 | 0.84 | 39 |
| Average/Total | 0.84 | 0.82 | 0.82 | 250 |

Table 4.7: Status Performance Classification Results

## 4.4 Summary

In this chapter, the results from the experiments using both the *Steam-PPA* and *AIR-SPC* systems have been presented. Some analysis within certain sections in this chapter had been performed to be able understand the results from other sections. A deeper analysis into these results is presented in the next chapter.

# Chapter 5

# Analysis

In this chapter, I analyze the results obtained from both the *Steam-PPA* and *AIR-SPC* system presented in Chapter 4. The emphasis is on presenting qualitative explanations of the figures, tables, and numbers presented earlier. In Section 5.1, I analyze the results from the status performance construction based on the acquired monetary values, as well as resulting categories from the cluster analysis. Section 5.2 covers my findings from analyzing tie-strength between players and users within the *Steam* social network. I also perform an in-depth analyses of each of the resultant meta-ties principal components which were acquired from PCA, describing what each represents and how it effectively provides an abstract way to reason about a player's social structures within a the *Steam* social network. Section 5.3 covers the results from *AIR-SPC*, where I discuss about the resultant model and its effectiveness for prediction and classification of status performance.

## 5.1 Status Performance

Here, I analyze the results from Section 4.1 of the previous chapter, obtained from constructing the performance status categories from the both players' TF2 hats equipped and inventory monetary values.

### 5.1.1 Equipped vs. Inventory Hat Value Distribution

In order to correlate the distribution of inventory hat values and equipped hat values data points across players, I made use of the Spearman's rank correlation coefficient. A correlation coefficient takes values in the range $[-1, +1]$, with the extreme values indicating that a perfect, monotone function exists between the two sets of values. Based on my dataset of 250 `(equipped, inventory)` status performance data points, there exists a high and positive correlation between the two variables (Spearman's $\rho = 0.67$, $p < 0.01$). Qualitatively, this correlation points towards the preference of players to have corresponding values of both equipped or inventory items. This means the chances of a player with a high equipped-value for his or her avatar, but a low inventory value, is low and unlikely to occur. Likewise, it is also unlikely that a player with a low inventory value would have a high equipped-value.

#### Player Behavior – Hoarding

However, an interesting observation is that almost no players had a **high equipped, low inventory** set of status performance values. This is highlighted in Figure 5-1. Looking at any other quadrants of the graph, only the highlighted region seems to be sparsely populated, indicating that players to achieve a high equipped value for status performance, a high inventory value is required. This could be a result of players who actively seek out the most expensive hats, inevitably replacing equipped high value hats with even higher value hats. As only one hat can be worn on a character at a time, this means that the previous high-value hat would be placed in the inventory. It also points towards a kind of **"hoarding"** behavior, since those players seem to never want to release expensive hats which they own, despite not being able to equip them on their characters.

### 5.1.2 Status Performance Cluster Analysis

From the cluster analysis performed in Section 4.1 of the Results chapter, I chose $k = 4$ as the number of cluster for the k-means algorithm. However, it was noted

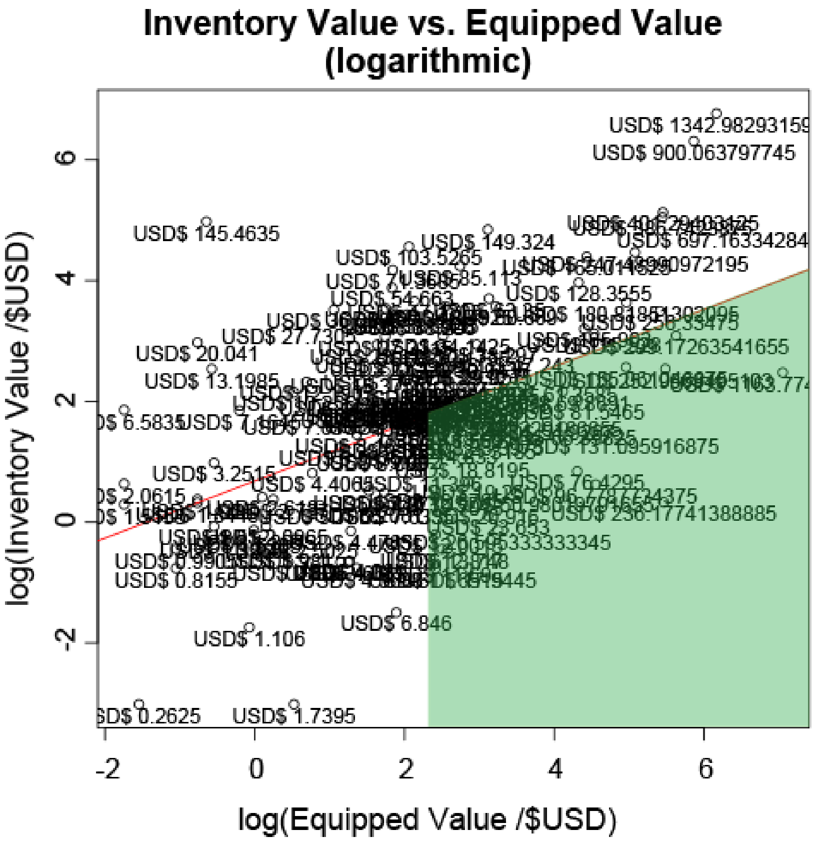


Figure 5-1: The scatter plot of status performance values of inventory against equipped values, but with the highlighted region showing little or no data-points.

that both the within-group sum of squares and the BIC scores for values of $k$ in the range $3 \leq k \leq 7$ were similar. As a result, I experimented with both increasing and decreasing increasing the number of clusters by 1, with values $k = 3$ and $k = 5$.

### Increasing the Number of Clusters ($k = 5$)

By increasing the number of clusters, I obtained a new set of mean values for our status performance data points, summarized in Table 5.1. The first thing I noted was that clusters at the extreme ends (`NONE` and `HIGH`) have similar means to those from before. The same ninety-one instances for the `NONE` status performance label have been clustered together. For the status performance cluster labeled `MEDIUM`, the mean values for both the equipped and inventory values are higher than before. The most interesting result lies in clusters 2 and 3, which both now form the categories of `LOW` status performance. This results in two different variations of LOW status performance, the first being a low status performance, but with a high equipped-to-inventory ratio (LOW-E), while the second being a low status performance, but with a low equipped-to-inventory ratio (LOW-I). This additional variance sheds light into possibly more of such gradients of status performance per status performance cluster (i.e., one might possibly have MEDIUM-E and MEDIUM-I, etc.), which would add more nuance to the categorization players according to status performance.

| | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 |
|---|---|---|---|---|---|
| Equipped | \$0.08 | \$7.00 | \$7.99 | \$56.32 | \$360.71 |
| Inventory | \$0.33 | \$3.82 | \$21.02 | \$38.25 | \$178.93 |
| Frequency | 91 | 100 | 24 | 24 | 11 |
| Status Performance | NONE | LOW-E | LOW-I | MEDIUM | HIGH |

Table 5.1: Status performance means with increased number of clusters ($k = 5$)

### Decreasing the Number of Clusters ($k = 3$)

By decreasing the number of clusters, I obtained a new set of mean values for the status performance data points, summarized in Table 5.2. From the table, I noted that the middle cluster (Cluster 2) has a mean value which would have likely made it categorized as LOW from our original results. From the frequencies outlining the distribution of players across the clusters, I note that the reduction of the number of clusters is less nuanced in its capacity to describe players' status performance. Quantitatively, the obtained BIC score ($BIC = -3251$) and log-likelihood value ($\ell = -1595$) for this clustering have a higher difference than those obtained by increasing the number of clusters. Overall, it appears that in order to capture a better model of the categories of players according to status performance, more clusters than $k = 3$ are required.

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| Equipped | \$0.25 | \$8.59 | \$154.32 |
| Inventory | \$0.49 | \$8.07 | \$85.23 |
| Frequency | 110 | 104 | 36 |
| Status Performance | `LOW` | `MEDIUM` | `HIGH` |

Table 5.2: Status performance means with reduced number of clusters ($k = 3$)

## 5.2 Tie Strength and Meta-Ties

In this section, analyze the obtained tie strength and meta-ties results from the Results chapter. First, I analyze the various tie strength predictive variables and their distribution over our dataset of players, with the aim of analyzing the correlation between predictive variables within each tie strength dimension with. Second, I focus on the results from constructing meta-ties from the PCA dimensionality reduction process. Being defined numerically using coefficients in Section 4.2, here I instead reason about what each meta-tie represents qualitatively. This provides a method of reasoning about the distribution of players in terms of variables and factors of the target domain, which in this case is *Steam* and TF2.

### 5.2.1 Tie Strength Predictive Variables in *Steam*

From the data collected from *Steam-PPA* used to calculate the tie strength predictive variables from the two-hundred fifty profiles, the first step was to take a closer look at each of the predictive variables' distribution, as well as performing a correlation test between predictive variables within the same tie strength dimension. In all the cases covered below, I once again made use of Spearman's rank correlation coefficient as the measure of correlation.

#### Intimacy Variables

Figure 5-2 shows the distribution curves and scatter-plots of the two variables, *Own Wall Posts* and *Friend Wall Posts*. From the distribution curves, it appears that both predictive variables have a skewed distribution across the collected profiles.

This points towards the activity of posting on walls as something either users fully engage in, or do not engage with at all. There was a moderately high and positive correlation between both predictive variables (Spearman's $\rho = 0.55$, $p < 0.01$)

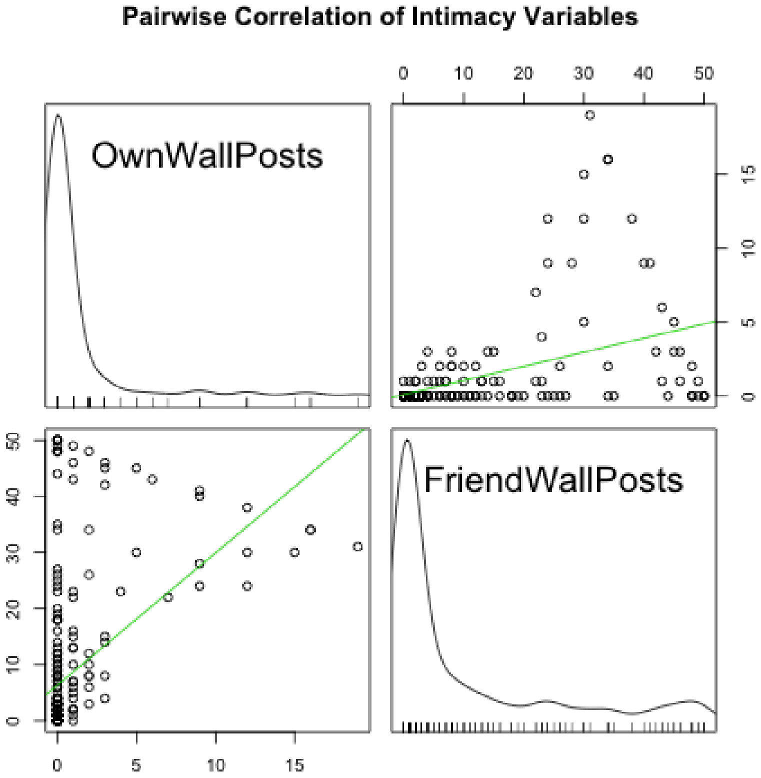


Figure 5-2: Pairwise scatter-plot of the Intimacy predictive variables.

### Intensity Variables

Figure 5-3 shows the distribution curves and scatter-plots of the two variables, *Friend Count* and *2$^{nd}$ Degree Friends*. From the distribution curves, it appears that both predictive variables have a varied distribution, with the distribution curve for the *2$^{nd}$ Degree Friends* being slightly more uniform. This indicates that the number of friends that each player had was never extremely low, and neither were a player's 2$^{\text{nd}}$ degree friend counts, pointing towards a kind of reinforcement of engagement between the player and his or her friends. There was a moderately high and positive correlation between both predictive variables (Spearman's $\rho = 0.58$, $p < 0.01$)

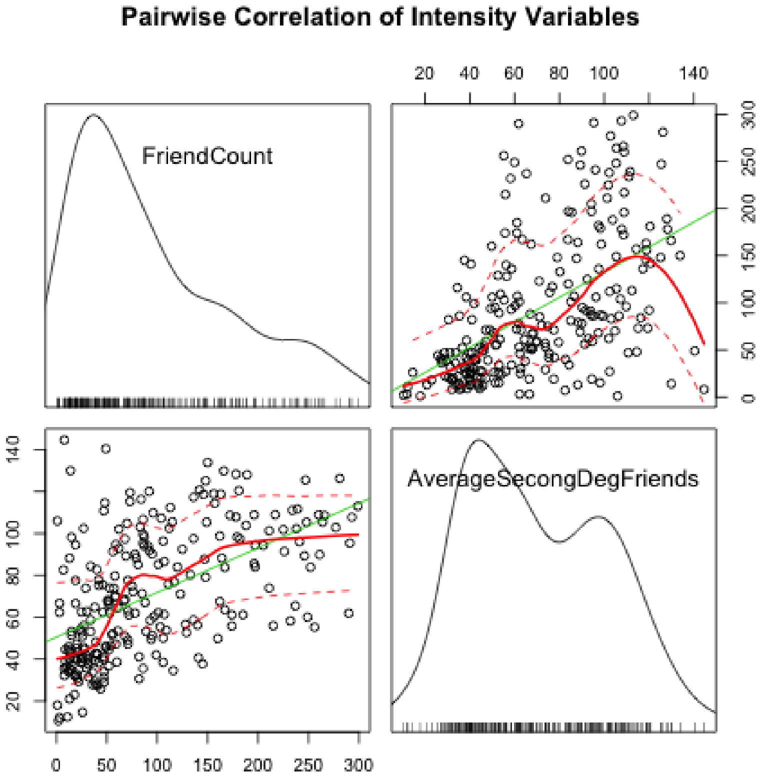


Figure 5-3: Pairwise scatter-plot of the Intimacy predictive variables.

### Reciprocal Services Variables

Figure 5-4 shows the distribution curves and scatter-plots of the two variables, *Average Mutual Friends* and *Average Mutual Groups*. From the distribution curves, both predictive variables have a slightly skewed distribution. It indicates that structurally, players also seemed to be relatively engaged in the system. There was not enough significance in our correlation test to draw a conclusion. Using the Pearson's Product-Moment Correlation test, there was a very low correlation between both predictive variables (Pearson's $r(25) = 0.15$, $p < 0.05$). This is somewhat expected since trading is application specific (TF2 in this case), and common applications would likely not factor into amount of traded items a player has.

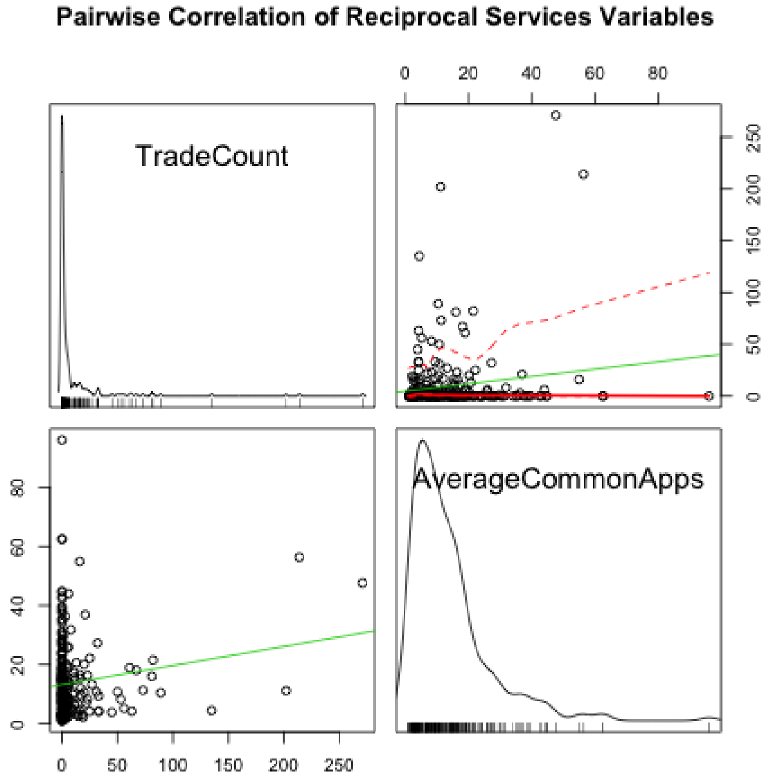


Figure 5-4: Pairwise scatter-plot of the Reciprocal Services predictive variables.

### Emotional Support Variables

Figure 5-5 shows the distribution curves and scatter-plots of the two variables, *Positive Emotion Words* and *Negative Emotion Words*. From the distribution curves, both predictive variables have highly skewed distributions. This is somewhat expected, since it was showed earlier that the distribution of both *Intimacy* predictive variables involving wall posts were skewed. There was a very high and positive correlation between both predictive variables (Spearman's $\rho = 0.86$, $p < 0.01$)

### Structural Variables

Figure 5-6 shows the distribution curves and scatter-plots of the two variables, *Average Mutual Friends* and *Average Mutual Groups*. From the distribution curves, both predictive variables have a slightly skewed distribution. It indicates that players had

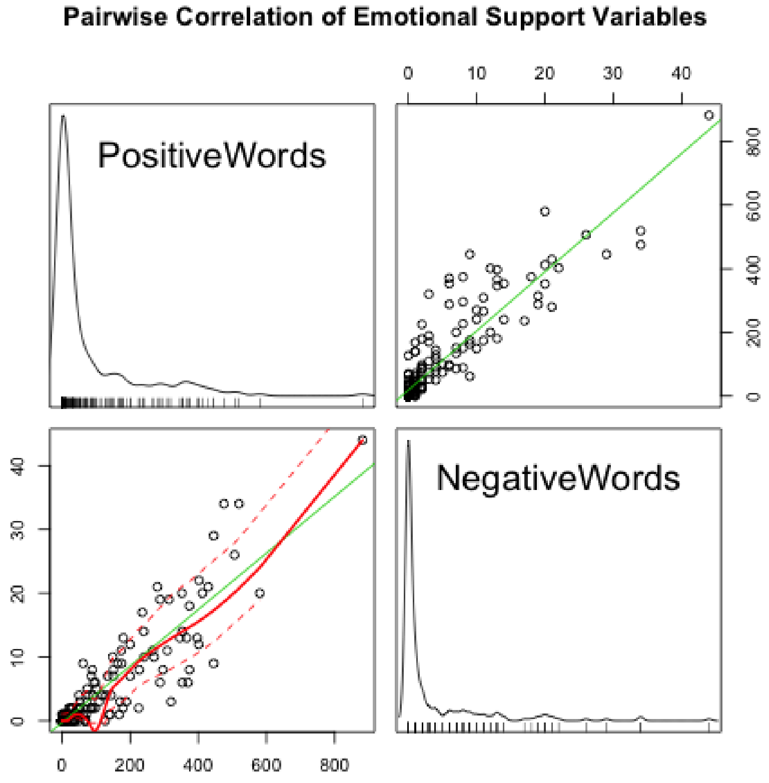


Figure 5-5: Pairwise scatter-plot of the Emotional Support predictive variables.

topologically dense networks. There was a high and positive correlation between both predictive variables (Spearman's $\rho = 0.61$, $p < 0.01$)

### 5.2.2 Interpreting Meta-ties

In Section 4.2.3 of the Results chapter, I presented the top five scoring meta-ties, which were the result of performing PCA on the dataset of player profiles, each represented as features of predictive variables. Each meta-tie was shown to be mathematically formulated as a weighted linear combination of the predictive variables. Here, I analyze each meta-tie qualitatively by studying the coefficients assigned to each predictive variable, and grouping them according to their signs (+/−). This approach allows us to reason about what each meta-tie represents in terms of the target domain, which in this case is the applications domains of TF2 and *Steam*, and

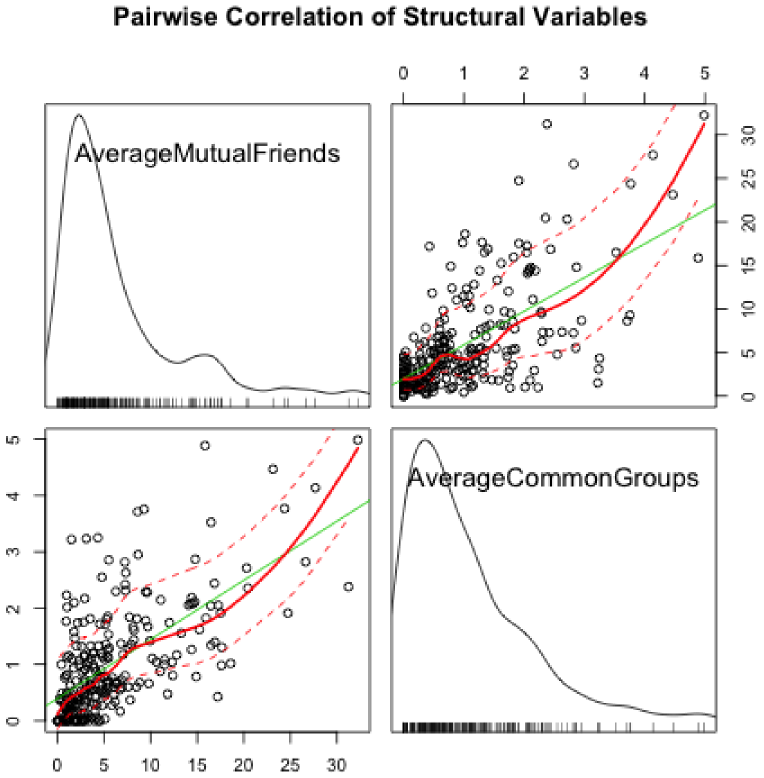


Figure 5-6: Pairwise scatter-plot of the Structural predictive variables.

allows one to gain insight into what factors of players' social structures separate them apart.

### Meta-tie #1: Longtime-Active/Presently-Active Index

Figure 5-7 shows the dataset of players projected over meta-tie #1 and meta-tie #2. Focusing first on meta-tie #1, it is observed that *Days as Friends*, a "Duration" predictive variable, is the only one one the positive axis. (We ignore *Common Applications* here, as its coefficient was 0). The rest of the predictive variables are projected along the negative axis. Meta-tie #1 appears to be a kind of **Longtime-Active/Presently-Active** index, describing players who have been long-time users of *Steam*, versus those who are active, but relatively newer users.

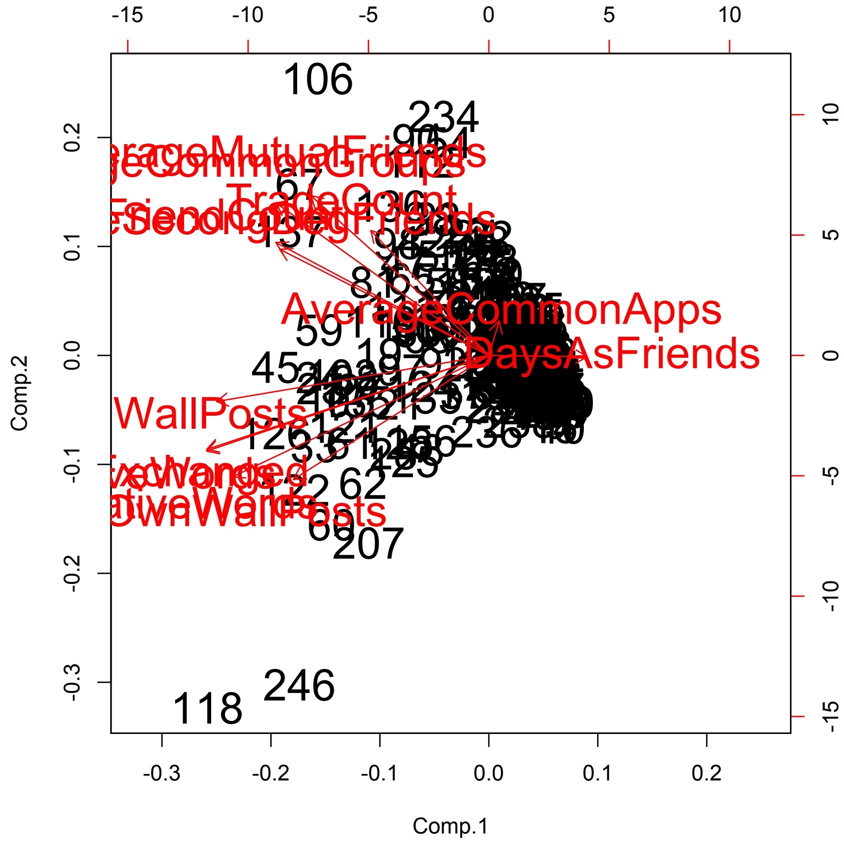


Figure 5-7: Projection of data onto meta-tie #1 and meta-tie #2.

### Meta-tie #2: Private/Public Social Interaction Index

Along the vertical axis of Figure 5-7, meta-tie #2 is analyzed. It is observed that there the predictive variables are divided into two groups, except for *Average Comon Applications* and *Days As Friends*, as listed below:

- Positive Vertical Axis
  - Trade Count (Reciprocal Services)
  - Mutual Friends (Structural)
  - Common Groups (Structural)
  - Friend Count (Intimacy)
  - 2$^{\text{nd}}$ Degree Friends (Intimacy)

- Negative Vertical Axis
  - Own Wall Posts (Intensity)
  - Friend Wall Posts (Intensity)
  - Words Exchanged (Intensity)
  - Positive Words (Emotional Support)
  - Negative Words (Emotional Support)

The first interesting observation is that there is no occurrence of a dimension of predictive variable being split across both axes – suggesting that the predictive variable dimensions are a robust categorization. The next step is to identify what each axis represents. The negative axis appears to consist of predictive variables related to public social interactions. The positive axis appears to consist of predictive variables related to network structure and links between members, which I term latent (or private) social interaction, which indicate that ties are present, but not necessarily exhibited or performed.

### Meta-tie #3: Familiar/Stranger Index

Figure 5-8 shows the dataset of players projected over meta-tie #1 and meta-tie #3. Along the vertical axis, meta-tie #3 is analyzed. It is observed that, once again, there the predictive variables are divided into two groups. Focusing on the predictive variables, which are clearly visible (corresponding to significant coefficient magnitudes), I obtained the two groups listed below:

- Positive Vertical Axis
  - Trade Count (Reciprocal Services)
  - Common Applications (Reciprocal Services)
  - Days as Friends (Duration)
- Negative Vertical Axis

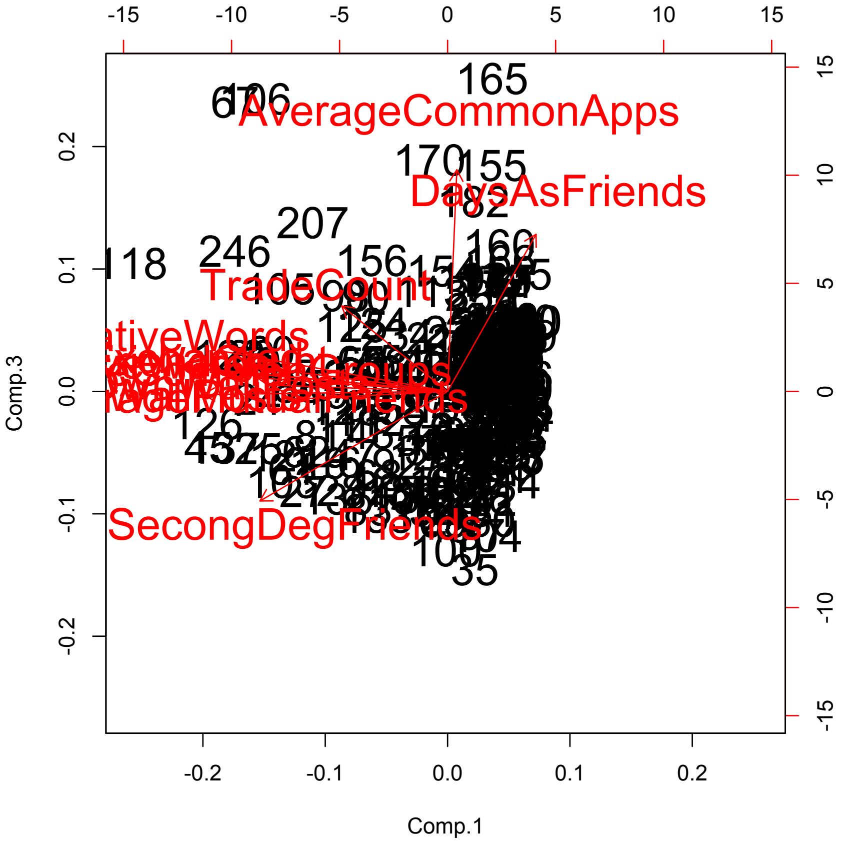


Figure 5-8: Projection of data onto meta-ties #1 and meta-tie #3.

– $2^{\text{nd}}$ Degree Friends (Intimacy)

The negative axis only contains the *$2^{nd}$ Degree Friends* predictive variable. The positive axis shows *Average Common Applications* and *Days as Friends* as the two predictive variables. Thus, meta-tie #3 distinguishes between players with large, $2^{\text{nd}}$ degree networks (friends of friends) versus players who have a common set of applications and have been friends for a long period of time. This points towards a kind of **Familiar/Stranger Index**.

### Meta-tie #4: Trader/Non-Trader Index

Figure 5-9 shows the dataset of players projected over meta-tie #1 and meta-tie #4. Along the vertical axis, meta-tie #4 is analyzed. It is observed that, by focusing on the predictive variables which are the most visible, that I may obtain the following categories:

- Positive Vertical Axis
  - Trade Count (Reciprocal Services)
  - Common Applications (Reciprocal Services)
- Negative Vertical Axis
  - Days as Friends (Duration)

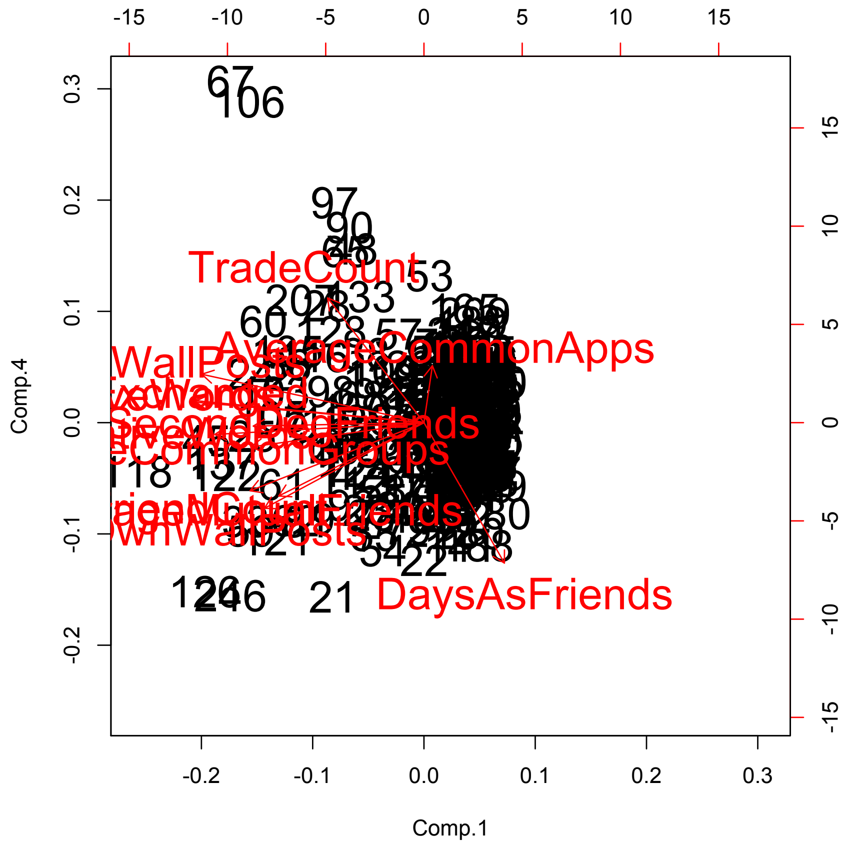


Figure 5-9: Projection of data onto meta-tie #1 and meta-tie #4.

The negative axis contains the *Days as Friends* predictive variable, indicating players who have friends whom they've known for a long while on *Steam*. The positive axis contains the *Trade Count* and *Common Applications* predictive variables, which points towards players with a high number of traded items. This meta-tie appears to separate players who trade actively and those who do not. Active traders are likely to possess shorter periods of knowing others on their friend lists as they are added solely for the purpose of trading, and likely even removing players due to the friend list limits imposed by *Steam*. Thus, this meta-tie is a **Trader/Non-Trader Index**.

### Meta-tie #5

Figure 5-10 shows the dataset of players projected over meta-tie #1 and meta-tie #5. Along the vertical axis, meta-tie #4 is analyzed. It is observed that, by focusing on the predictive variables that are the most visible, that following categories were obtained:

- Positive Vertical Axis
    - Trade Count (Reciprocal Services)
    - Days as Friends (Duration)
- Negative Vertical Axis
    - Common Applications (Reciprocal Services)
    - Mutual Friends (Structural)
    - Common Groups (Structural)

The negative axis contains the three predictive variables *Common Applications*, *Mutual Friends*, and *Common Groups*. These three predictive variables share the same property in possessing a "common" factor (mutual friends is technically calculated as common friends), reflecting players who are exhibit social characteristics. The positive axis consists of the predictive variables *Trade Count* and *Days as Friends*, which identifies players who are some engaged in either *Steam* (friendship duration) or TF2 (involvement in trading) for a functional purpose. This meta-tie appears to be a kind of **Functional/Social Index**.

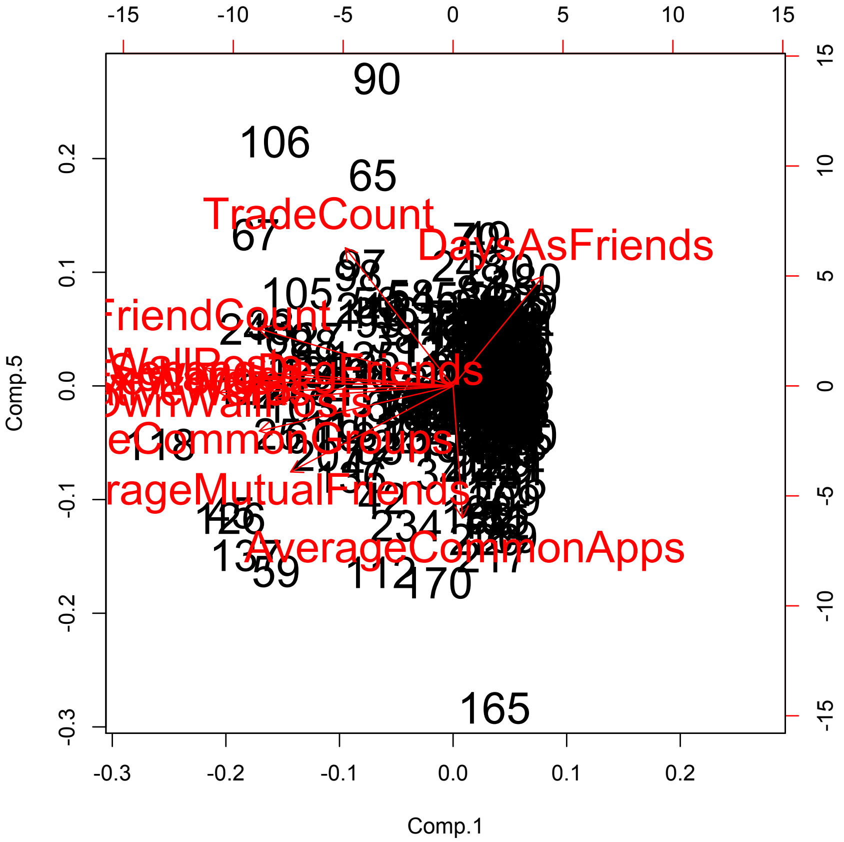


Figure 5-10: Projection of data onto meta-tie #1 and meta-tie #5.

## 5.3 Classifying Status Performance with Meta-ties

In this section, I analyze the results from associating the status performance clusters with the meta-ties clusters from Section 4.3.2 of the Results chapter. First, I study the degree of association between the two clusters in order to see if there are any correspondence between meta-ties (from the social network) and status performance (from in-game). Second, I analyze the results of the trained SVM model and its classification performance.

### 5.3.1 Degree of Association between Clusters

To analyze any association between our two variables – status performance and meta-ties; the test performed was the Cramér's V measure of association, based on the inter-correlation between our variables. I observed that there is a moderate association between both clusters (Cramér's V $_c$ = 0.449). This indicates that when categorizing players categorized into clusters for both meta-ties, and for status performance, that a degree of relationship exists between both variables, and added weight into studying the relationship between both status performance and meta-ties in more detail, particularly if any causal relationship exists between them.

### 5.3.2 Classification Performance with SVMs

Here, the results from Section 4.3.3 are analyzed, covering the performance of the trained SVM model. First, I analyze its performance in "fitting" the data – in which the model is trained and tested on the same data. Second, I analyze results from having separate training and validation sets of data.

#### Fitted Model Performance

First, it is noted that for `MEDIUM` status performance, the model was fitted perfectly. Secondly, for `HIGH` status performance, the model had a perfect precision, but a recall of 72%, which suggests that the model serves as a useful `HIGH` status performance discriminant, but less well as an extractor. It would thus be useful for a scenario where, given a sample of a population of users (or players) which we know contains high status performance individuals, we could use the model to identify them accurately. It less useful if one is unsure about whether a population of players contains any `HIGH` status performance individuals at all, though about 72% likely to pick the correct ones.

Next, I observed that both `NONE` and `LOW` status performance categories have exactly the same F-1 scores. The difference lies in that the model has better precision in classifying `LOW` status performance, but a higher recall for `NONE` status performance. Since in both status performance categories, there is a comparable number of samples, this indicates that the model is in general useful for identifying `NONE` and `LOW` categories of players. Given a scenario where one knows that a sample of the population of users have NONE status performance (e.g., idle users, non-spenders, etc.), one could identify them accurately. However, if one is unsure that such players exist at all, he or she should likely choose to identify those with LOW status performance due to its better performance in such a scenario.

### Classification Performance

The achieved accuracy of the model was 58% with separate training and validation sets of data (repeatedly validated using 4-fold cross-validation). This is a significant result because of the fact that we have 4 classification labels of status performance, meaning that the model has a significantly better performance than random guesses. Additionally, the status performance clusters had uneven distributions, which also adds to the difficulty of this classification problem (which I tried to mitigate using stratified cross-validations). Perhaps most significantly, the set up of the classification problem involved two application domains of a social network, and in-game performance, which are fairly separate on a technical and semantic level. Hence, this result shows promise in the effectiveness of the methods employed and the chosen domain.

## 5.4 Summary

In this chapter, I have analyzed the results obtained from both the *Steam-PPA* and *AIR-SPC* systems for the experiments conducted. This consisted of quantitatively analyzing, as well as providing a qualitative assessment of the numerical results. In some cases, an informed reasoning of the analysis was applied, which provided broader insight into the characteristics of players' social network and performance. In the next chapter, I discuss implications of these findings, and conclude with proposed extensions to improve upon this work.

# Chapter 6

# Conclusion

In this chapter, I conclude with an overview of the work undertaken in this thesis, touching on the various experimental results and take-aways based on the analysis and discussions from the previous chapters. I also discuss about the limitations of the system, together with potential avenues for extending the research future work. Section 6.1 discusses the potential implications of these results. Section 6.2 covers the concluding reflections based on the work undertaken in this thesis, together with its contributions. Section 6.3 discusses ways to improve both the systems and methods used, and suggests ways to extend upon the work of this research. Section 6.4 ends the chapter, and this thesis, with closing remarks.

## 6.1 Implications of Findings

In this section, I discuss about the potential implications of these results. First, I cover potential implications related to game design (avatar customization, constructions of virtual economies), and focus particularly on the integration of social networking and gaming platforms. Next, the discussion shifts to focus on issues related to people and societies in the real-world, relating such social issues back to user/player identity construction in computational technologies

### 6.1.1 Game Design Implications

Games incorporating avatar customization systems should consider how customization choices available to the players go beyond an item's aesthetic appearance. This is important as players derive an item's value based on other factors, such as its potential for self-expression. The choices to purchase, earn, or otherwise acquire these items are based not only on their visual appearances or functional benefits within the game world. Users' choices in customizing their avatars using items can also be a reflection of the player's real world identity. That is, player preferences for avatar items can be seen as a performance, not just within the game world, but also with influences from real-world notions of taste, social structures, and cultural values. Thus, implementing platforms that integrate both real-world and virtual-world identities, developers should consider the effects of doing so beyond mere information transfer from one domain to the other for purposes like finding teams for enaging in virtual combat.

### 6.1.2 Social Implications

It is also important to consider implications related to social issues that might arise out of computational identity representation systems, especially with the high levels of interaction that occur between players, as well as with developers. Inference regarding a player's real-world identity and preferences can be correlated with their behaviors in virtual worlds including avatar creation and customization (and vice versa). Creating items for distribution in a virtual environment has similarities to the construction of value for real-world items. Looking at hats in TF2 based upon factors such as mode of acquisition, promotions by developers, monetary value, and so on parallels real world phenomena, such as the appeal of designer or limited edition goods. One can examine the different categories of people who seek to acquire particular virtual items or classes of virtual items (e.g., people with the means to seek out expensive items, people who care about aesthetics, and so on) and predict how they might perform status in a gaming/virtual world. In constructing virtual economies, consideration of social effects must go beyond enhancing or balancing gameplay and should include sociological issues such as privilege and marginalization.

## 6.2 Concluding Reflections

In this thesis, I have presented a work based upon the AIR Model, a cognitively-grounded approach to representing identity through computational technologies. The motivation for such an approach is to develop more robust technologies in order to avoid the limitations of existing computational identity representation systems that fail to adequately provide users with the capabilities to represent themselves in digital environments. Such limitations have the effect of reinforcing undesirable structures that exist in the real-world into the virtual world. Both systems developed in this thesis seek to understand the factors involved in representing one's real-world identity computationally within virtual environments, spanning two different domains—a social network, and a videogame avatar.

The first system, the *Steam-Player-Preference Analyzer (Steam-PPA)*, collects publicly available information from players' social networking profile on the *Steam* network, and from the commercially successful multi-player online videogame *Team Fortress 2* (TF2). The second system, the *AIR Toolkit Status Performance Classifier (AIR-SPC)*, uses machine learning techniques to create a model of the data and can be used to predict a players' preferences (or performance) in TF2 using information derived from his or her social network. I introduced and defined **status performance** as a computational representation of a player's preference in performance in TF2, related to the real-world monetary value of the virtual item hats, which are used to customized one's avatar. I showed a strong correlation between the value of a player's used and collected hats, and illustrated the effects of clustering to divide the player space into separate categories of status performance.

With the *Steam-PPA* system, I have presented an approach to obtaining variables used for estimating tie-strength in the social network Steam. This work also suggests Principal Component Analysis (PCA) as an effective technique to reduce the dimensionality of tie strength variables into a smaller, abstract set of social value principal components that still describe the original dataset, termed **meta-ties**. With information from two different domains (a social network and a game), we showcase the effectiveness of using Support Vector Machines (SVMs) in learning to classify status performances using meta-ties. This main result of the paper highlights the existence of a strong relationship between a player's real-world identity and virtual identity within games. My hope is to motivate designers of computational identity systems in games and social networks to consider the importance of providing adequate technologies for users of such systems, and to remember to consider the effects of any coupling between real-world and virtual identities, through games, social networks, and most prominently integrated hybrids of both.

## 6.3 Limitations and Future Work

In this section, I discuss some of the areas of this work which could be improved, and consequentially outline several ways to extend this work for future work.

### 6.3.1 Increasing the Sample Size of Analyzed Profiles

In the process of obtaining the results we have presented, I discovered significantly better classification performance by the *AIR-SPC* system when increasing the number of profiles analyzed from one-hundred fifty to two-hundred fifty, with the overall F-1 Score increasing from 69% to 82%. Based on the analysis of the status performance clusters in Section 5.1.2 of the Analysis chapter, the increase in the number of clusters adds nuance to the categorization of players. I suspect that the additional data improved the robustness of the *AIR-SPC* system. Consequentially, the additional data might contribute towards making the distribution between status performance clusters more even, as currently, the lower status performance clusters (`NONE` and `LOW`) have more samples than the higher status performance clusters (`MEDIUM` and `HIGH`).

### 6.3.2 Increasing the Number of Tie Strength Predictive Variables

Another improvement that was observed was increasing the number of predictive variables, used to estimate tie strength in *Steam*, from ten to twelve. Extensions to *Steam-PPA* in order to encompass more of the predictive variables presented by Gilbert and Karahalios [19] would theoretically provide more insight into the social structures of the players. With *Steam* continuously improving as a social networking service (with added features like photo-sharing and commenting), it would be interesting even just to compare the effects of the predictive variables against a more popular, less gamer-oriented social network like Facebook or Twitter.

### 6.3.3 Additional Database for Sentiment Analysis

Performing sentiment analysis using an additional database, using some kind of hierarchical classification, could theoretically improve the sentiment analysis performance of *AIR-SPC*. An additional database providing word-emotion data is the Linguistic Inquiry and Word Count (LIWC) database for categorizing word emotions, which is often used in the field of human-computer interaction (HCI).

### 6.3.4 Gaining More Insight into In-game Player Behavior

Some preliminary work has been undertaken into using *Steam-PPA* to collect publicly available data which focus more on players' in-game behaviors, such as favorite character class (total character class played time) and most effective character class (best scoring character class). Analyzing gameplay-related statistics would provide an additional lens into identity representation as performed by the user. For example, one could use *Steam-PPA* to calculate the class of the player's most expensively clad character player's and correlate it to either the player's favorite or most effectively played class. This would also include other aspects such as trophies and achievements within the game world.

### 6.3.5 Reversed Classification

An interesting extension would be to investigate whether social values can be predicted from status performance for players. This reverse classification would provide insight into how gameplay in the virtual world translates into the world of social networking (which many users consider to be close to the real world). It would allow us to better study the relationship between identity and behavior in-games and in relationships on social networks.

## 6.4 Closing Remarks

There are aspects associated with the creation and representation of one's real-world identity computationally that should be considered when developing technologies which support either. This is even more true when a system integrates information from both. One's performance with a virtual environment, even within a fantastical or comical setting, can be correlated with the social structures in the physical world. As such, there is great potential for system developers to implement technical infrastructures that can adequately support a user and his or her self-expression through computational mediums. Such systems can be more expressive for users and can avoid, or even combat, the reinforcement of disempowering social identity issues in real, virtual, and hybrid worlds.

# Bibliography

[1] Mark Ackerman. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction*, 15(2):179–203, September 2000.

[2] Roi Becker, Yifat Chernihov, Yuval Shavitt, and Noa Zilberman. An analysis of the Steam community network evolution. *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, pages 1–5, November 2012.

[3] James C. Bezdek, Thomas R. Reichherzer, Gek Sok Lim, and Yianni Attikiouzel. Multiple-prototype classifier design. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(1):67–79, 1998.

[4] Nick Bostrom. The Future of Identity. *Commisioned Report of The United Kingdom's Government Office for Science*, 2011.

[5] Geoffrey C. Bowker and Susan Leigh Star. *Sorting things out: Classification and its consequences.* The MIT Press, 2000.

[6] Ronald S. Burt. *Structural holes: The social structure of competition.* Harvard University Press, 2009.

[7] Roberto Cordeschi. *The discovery of the artificial: Behavior, mind and machines before and beyond cybernetics*, volume 28. Kluwer Academic Pub, 2002.

[8] Roberto Cordeschi. Steps towards the synthetic method. symbolic information processing and self-organizing systems in early artificial intelligence modeling. *The Mechanical Mind in History*, pages 219–258, 2008.

[9] Julian Dibbell. A Rape in Cyberspace; or How an Evil Clown, a Haitian Trickster Spirit, Two Wizards, and a Cast of Dozens Turned a Database into a Society. *High Noon on the Electronic Frontier: Conceptual Issues in Cyberspace*, page 375, 1996.

[10] Anders Drachen, Rafet Sifa, Christian Bauckhage, and Christian Thurau. Guns, Swords and Data: Clustering of Player Behavior in Computer Games in the Wild. *Proceedings of IEEE Conference on Computational Intelligence in Games*, pages 163–170, 2012.

[11] Emilio Ferrara, Pasquale De Meo, Giacomo Fiumara, and Alessandro Provetti. The role of strong and weak ties in Facebook : a community structure perspective. *Proceedings of International Conference of Computer Science ICCS*, pages 1–10, 2012.

[12] Marcello Frixione and Antonio Lieto. Representing Concepts in Artificial Systems: A Clash of Requirements. *Proceedings in the 4th Workshop on Human Centered Processes (HCP)*, February 2011.

[13] Francesco Gagliardi. A Prototype-Exemplars Hybrid Cognitive Model of " Phenomenon of Typicality " in Categorization : A Case Study in Biological Classification Theories of Categorization. 2005.

[14] Francesco Gagliardi. The Need of an Interdisciplinary Approach based on Computational Modelling in the Study of Categorization . Categorization in Cognitive Psychology and the Prototypes-Exemplars Debate Machine Learning and. pages 442–443, 2005.

[15] Francesco Gagliardi. The Necessity of Machine Learning and Epistemology in the Development of Categorization Theories : A Case Study in Prototype-Exemplar Debate. pages 182–191, 2009.

[16] Peter Gärdenfors. *Conceptual spaces: The geometry of thought.* MIT press, 2004.

[17] James Paul Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1):20, October 2003.

[18] Ben Geisler. Integrated machine learning for behavior modeling in video games. In *Challenges in game artificial intelligence: papers from the 2004 AAAI workshop. AAAI Press, Menlo Park*, pages 54–62, 2004.

[19] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, page 211, 2009.

[20] Erving Goffman. *The presentation of self in everyday life.* New York: Anchor Books, 1959.

[21] Joseph A. Goguen. Toward a social, ethical theory of information. *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide*, page 27, 1997.

[22] Joseph A. Goguen. Against technological determinism. *CVS*, page 87, 2003.

[23] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[24] Jonathan Grudin. Computer-supported cooperative work: History and focus. *Computer*, 27(5):19–26, 1994.

[25] D. Fox Harrell. Digital Metaphors for Phantom Selves: Computation, Mathematics, and Identity in Speculative and Fantastic Fiction and Gaming. *The Sublime in the Fantastic: The 29th International Conference on the Fantastic in the Arts*, 2008.

[26] D. Fox Harrell. Computational and Cognitive Infrastructures of Stigma : Empowering Identity in Social Computing and Gaming. pages 49–58, 2009.

[27] D. Fox Harrell. Designing empowering and critical identities in social computing and gaming. *CoDesign: International Journal of CoCreation in Design and the Arts*, 6(4):187–206, December 2010.

[28] D. Fox Harrell. Toward a theory of critical computing: The case of social identity representation in digital media applications. *CTheory*, 2010.

[29] D Fox Harrell. *Phantasmal Media: An Approach to Imagination, Computation, and Expression.* MIT Press, in press. edition, 2013.

[30] D. Fox Harrell, Greg Vargas, and Rebecca Perry. Steps Toward the AIR Toolkit: An Approach to Modeling Social Identity Phenomena in Computational Media. In *Proceedings of the 2nd International Conference on Computational Creativity*, 2011.

[31] Roberto R. Heredia and Jeffrey M. Brown. Code switching. `http://www.tamiu.edu/~rheredia/switch.htm`.

[32] Sotiris B. Kotsiantis, Ioannis D. Zaharakis, and Panagiotis E. Pintelas. Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, 160:3, 2007.

[33] Ludmila I. Kuncheva and James C. Bezdek. Nearest prototype classification: Clustering, genetic algorithms, or random search? *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(1):160–164, 1998.

[34] George Lakoff. *Women , Fire , and Dangerous Things.* University of Chicago Press, 1987.

[35] Chong-U Lim and D. Fox Harrell. Modeling Player Preferences in Avatar Customization Using Social Network Data: A Case-Study Using Virtual Items in Team Fortress 2. *Proceedings of IEEE Conference on Computational Intelligence in Games*, 2013.

[36] Nan Lin, Walter M. Ensel, and John C. Vaughn. Social resources and strength of ties: Structural factors in occupational status attainment. *American sociological review*, pages 393–405, 1981.

[37] Hugo Liu. Social Network Profiles as Taste Performances. *Journal of Computer-Mediated Communication*, 13(1):252–275, October 2007.

[38] Marlos C. Machado, Gisele L. Pappa, and Luiz Chaimowicz. A binary classification approach for automatic preference modeling of virtual agents in civilization iv. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 155–162. IEEE, 2012.

[39] David J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[40] Paul Manwaring. The $50 million virtual millinery. `http://theonlinesociety.com/2011/12/the-50-million-virtual-millinery/`, Dec 2011.

[41] Josh McCoy, Mike Treanor, Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. Prom week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 319–321. ACM, 2011.

[42] Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. Authoring Game-based Interactive Narrative using Social Games and Comme il Faut. In *Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate*, 2010.

[43] Douglas L. Medin and Marguerite M. Schaffer. Context theory of classification learning. *Psychological review*, 85(3):207, 1978.

[44] Jason Mitchell, Moby Francke, and Dhabih Eng. Illustrative Rendering in Team Fortress 2. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 71–76. ACM, 2007.

[45] Christopher Moore. Hats of affect: a study of affect, achievements and hats in Team Fortress 2. *Game Studies*, 11(1):1–28, 2011.

[46] Yi Mou and Wei Peng. Gender and Racial Stereotypes in Popular Video Games. *Handbook of Research on Effective Electronic Gaming in Education*, pages 922–937, 2008.

[47] Gregory L Murphy. *The big book of concepts*. The MIT Press, 2004.

[48] Gregory L Murphy, Douglas L Medin, et al. The role of theories in conceptual coherence. *Psychological review*, 92(3):289–316, 1985.

[49] Edwina L. Rissland. AI and similarity. *Intelligent Systems, IEEE*, 21(3):39–49, 2006.

[50] Eleanor Rosch. Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3):192, 1975.

[51] Eleanor Rosch. Principles of Categorization. *Foundations of Cognitive Psychology: Core Readings*, page 251, 2002.

[52] Eleanor Rosch and Carolyn B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.

[53] Doris C. Rusch and Matthew J. Weise. Games about love and trust?: harnessing the power of metaphors for experience design. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, pages 89–97. ACM, 2008.

[54] David B Skalak. Using a Genetic Algorithm to Learn Prototypes for Case Retrieval and Classification. *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning*, pages 64–69, 1993.

[55] Ruck Thawonmas and Masayoshi Kurashige. Clustering of online game users based on their trails using self-organizing map. *International Journal of Computer Games Technology*, 2008.

[56] Barbara Tversky and Kathleen Hemenway. Objects, parts, and categories. *Journal of experimental psychology: General*, 113(2):169, 1984.

[57] Valve. Team fortress 2 official wiki - classes. `http://wiki.teamfortress.com/wiki/classes`, Febuary 2013.

[58] Gregory Vargas. A Cognitive Categorization-Based Approach for Understanding Identity Representation Online. Master's thesis, MIT, 2010.

[59] Yanis Varoufakis. Arbitration and equilibrium in the team fortress 2 economy. `http://blogs.valvesoftware.com/economics/`, jun 2012.

[60] Barry Wellman and Scot Wortley. Different strokes from different folks: Community ties and social support. *American journal of Sociology*, pages 558–588, 1990.

[61] Dmitri Williams, Nicole Martins, Mia Consalvo, and James D. Ivory. The virtual census: Representations of gender, race and age in video games. *New Media & Society*, 11(5):815–834, 2009.