SOCIAL-EXPERT: AN EXPERTISE RATING ALGORITHM

By

Kawsar Kamal

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

_____

Koushik Kar, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

April 2010
(For Graduation May 2010)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to thank my professor and thesis advisor Dr. Koushik Kar for inspiring me to study the field of networking. I am very grateful to him for providing me with continuous guidance and intellectual challenge throughout this work.

Thanks also to my parents Mustafa Kamal and Rokeya Sharif for instilling the value of hard work and making opportunities available for me at every stage of life. Finally, a special thanks to my wife Naureen Akhter for her support and understanding. She deserves a degree for putting up with me while I completed this thesis.

ABSTRACT

We study the problem of expert identification in organizations and online communities. The process of finding experts in real life often relies on using networking in the form of word-of-mouth referrals. With this intuition we design a novel expertise finding algorithm called Social-expert that uses social networks. We define social networks that have a system of explicit expertise evaluation or feedback among members as: expertise networks. For each individual in an expertise network, social-expert considers two qualities: peer-rating and credibility. Peer-rating reflects someone's expertise on a topic while credibility represents his or her aptitude in accurately evaluating others' expertise on that topic. Social-expert calculates a global peer-rating and credibility score for each person based on peer expertise evaluations received and made by that person. We test the social-expert algorithm for convergence and effectiveness on a series of expertise network graphs which were constructed from simulated and real world data. Our results show that social-expert outputs converge to stable values relatively quickly and that the effectiveness is dependent upon a few factors. We find that social-expert can effectively identify nodes' expertise and credibility, given most evaluations in the network are somewhat reasonable. We provide some arguments for why this is likely to be the case for real life expertise networks in academic, professional or online communities. Another factor in social-expert effectiveness is how connected the graph is. Based on social-expert design, the more evaluations a node receives or makes, the more accurate we expect his or her peer-rating and credibility scores to be. Consistent with this expectation, our results show that social-expert was more effective in expertise networks with a higher average number of evaluations per node.

CHAPTER 1
Introduction

Alice is a Java programmer who is tasked with implementing a new feature for a web application. She comes across a few different open source frameworks that can potentially make her task easier. After some online searching she cannot decide which framework is most suitable, or whether to use a framework at all. Ideally she would like to consult with someone who can advise her on the frameworks with respect to code quality, maintenance, time to implement, learning curve, testing, packaging, licensing issues and so on. At this time Alice can reach out to her immediate work colleagues to find a suitable person. If her immediate colleagues do not have the pertinent background, they can refer her to other individuals who may be able to help.

Like Alice, almost everyone is occasionally faced with the problem of finding an expert. Often this is solved through networking; an expert is eventually found through word-of-mouth referrals. In this work we use contribute to the area of expertise finding. Expertise identification systems have been studied typically for implementation in professional organizations and online communities.

In this work we consider "expertise" to encompass the qualities of skill, experience, talent and knowledge. An "expert" is someone who has the minimum amount of expertise on a specific topic, to help a seeker on that topic. We adopt these definitions from a five-month long study [1] of expertise location in a real software firm. The study finds that the process of "expertise identification" is very different than that of "expertise selection". Expertise selection is concerned with how the seeker chooses the person to ask for advice once some suitable candidates are already available. Our work is concerned with expertise identification, that is, to find suitable candidates who possess a minimum level of expertise on a desired topic.

We propose a new algorithm called social-expert that uses social networks to find experts. A social network represents a set of individuals that are linked to one

another. The degree of linkage varies among individuals, that is, some have more links than others. The type of links or edges in social networks can be very extensive [2]. We define a special type of social network that we call *expertise network*, where the edges represent explicit peer evaluations or feedback. For example, Alice may choose to rate Bob as having 4 out of 5 expertise level on a certain topic. In that case we would place an edge from Alice to Bob with a weight of 4. All edges of an expertise network represent peer evaluations on the same topic. We expect that such expertise networks can be implemented in academic or professional settings, or in online communities. Evaluations may be collected in form of feedback after mutual interactions. The Advogato network [3] is an example of an online community that supports explicit peer evaluation among members. In this work we focus on expert identification once an expertise network is already available. A system of collecting peer evaluation data in organizations is outside the scope of this work; we suggest this as a future study. To test the social-expert algorithm comprehensively we construct a set of expertise network graphs with different characteristics. We obtain the underlying data for these graphs from simulation and real world datasets. Simulations allow us to test social-expert with variations in the distribution of nodes' expertise and credibility. Using datasets from existing online networks means that we can preserve structural properties[1] of naturally occurring online social networks.

The social-expert algorithm considers two qualities for each person in an expertise network: peer-rating and credibility. Peer-rating reflects someone's expertise on a topic as perceived by the network. Credibility represents his or her aptitude in accurately evaluating others' expertise on that topic. The final output of social-expert is a global peer-rating and credibility score for each person. Social expert calculates peer-rating for someone from a weighted average of peer evaluations received by him or her. The weights are the credibility scores, which are calculated as a function of the difference of opinion between the evaluator and the remaining network.

Our results show that social-expert outputs converge to stable values relatively quickly. The output values become progressively stable to an increasing number of

[1]These special properties include: degree of linkage variation, clustering and small diameter. We discuss them further in section 4.1.3 with appropriate background.

decimal places: within 30 iterations most nodes' scores are stable to 17 decimal places. To calculate peer-rating and credibility values at any iteration, social-expert uses the values calculated at the previous iteration. Therefore, before the first iteration can take place, we must initialize each node in the graph with some initial peer-rating and credibility estimates. Our results show that the final social-expert output values are not dependent upon these estimates. As well as convergence, we measure the effectiveness of social-expert in estimating the true expertise and credibility of nodes. We observe that the level of effectiveness is dependent upon a few factors. One important factor is the accuracy of peer evaluations: as long as about 80% of evaluations in the network are somewhat reasonable, social-expert can effectively identify nodes' expertise and credibility. We provide some arguments for why this is likely to be the case for real life expertise networks in academic, professional or online communities. Another factor in social-expert effectiveness is how connected the graph is. We show that in general social-expert is more effective in expertise networks with a higher average number of evaluations per node. Based on social-expert design, the more evaluations a node receives and makes, the more accurate we expect his or her peer-rating and credibility scores to be. Consistent with this expectation our results show that social-expert is not as effective for nodes that receive or make only one or two evaluations and each evaluation is inaccurate. We propose an improvement of adding a confidence value to each node's peer-rating and credibility score based on how many evaluations the node receives or makes.

We begin our work in this chapter by providing some context and motivation for expertise identification and the social-expert algorithm. The rest of this thesis is organized as follows. In chapter 2, we introduce existing works and formalize our notion of an expertise network. We then describe the social-expert algorithm in chapter 3, along with motivations for its design. In chapter 4, we describe the process of building expertise network graphs, followed by our experiment design. Chapter 5 outlines important results and findings from testing social-expert. We evaluate our findings, propose improvements and future work in chapter 6.

CHAPTER 2
Background and Related Work

At its essence social-expert is an algorithm that takes a network graph as its input and calculates a score for each node with the purpose of producing meaningful rankings. Such graph based ranking algorithms have been an active area of research for some time. Two very well known works include Google's Pagerank [4] and Kleinberg's HITS algorithm [5]. Both of these use the web's hyper-link structure to rank websites. Aside from websites, graph algorithms may be designed to operate on various types of networks including P2P [6, 7], mobile device networks [8], e-mail networks [9] and so on. We are specifically researching a ranking algorithm for graphs that represent online (or offline) social networks[1] among people. There have been a series of works studying social networks in areas such as trust, recommender systems, reputation and expertise. Sabater and Sierra in their work [10] present a comprehensive survey on algorithms that compute trust and reputation in social networks. Golbeck in her PhD. thesis [11] presents a set of algorithms for inferring trust in social networks. More examples include graph algorithms such as SocialTrust [12], Advogato trust metric [13] and others.

Our contribution is in the area of expertise location in social networks. By extracting information inherent in social networks we want to find some individuals that are experts on a given topic. Expertise location or finder systems have been studied in a series of works which can be broadly classified as those that use social networks and those that rely on other information and heuristic methods.

Who Knows [14] is an example of an early expertise location system that does not use a social network. It uses a profile matching mechanism where user profiles are created by text indexing submitted works of each individual. A query by a seeker on a particular topic would then produce a list of experts whose profiles best matched the query [15]. Similarly, the "Expert finder" system [16] by Vivacqua and Lieberman performs automatic profiling of Java code and calculates an expertise

[1] Although the term "social networks" often refers to recreational friendship networks, we are referring to groups of people linked by academic, professional or organizational activities.

score for each user in various Java packages. Another work with the same title, "Expert finder" [17] by Mattox et al., also determines expertise scores for users on various topics based on published documents, resume and other sources. The ContactFinder agent [18] parses bulletin board messages to extract key contacts for topic areas. It then replies to board questions with referrals of best matched individuals for the topic at hand [18]. The above systems in general perform expertise identification by matching a query to individuals with the highest score or most relevant profile for the topic being queried.

Examples of systems that use other methods include the Answer Garden 2 [19], proposed by McDonald and Ackerman. In this system an autonomous "escalation agent" forwards the query to a live chat channel which comprises of users in the same work group or hallway (or other type of grouping) as the seeker. If there are no experts in the chat channel for the relevant topic, the escalation agent may rely on other resources such as a live helpdesk [19]. The authors of Answer Garden 2, in their 1998 work [1] performed an important field study of expertise identification in a real software company. The heuristics found in this study were then encapsulated and proposed in the comprehensive Expertise Recommender (ER) [20] system. The ER architecture features pluggable modules for expert identification using heuristics such as a file's change history, or searching support databases for individuals that have solved problems in this topic before.

An important aspect of expertise identification discovered in the 1998 study [1] was that there were some individuals with "*very strong very elaborated social networks*" [1]. These individuals fulfilled an "Expertise Concierge" role within the organization. As well as possessing high technical competency, they were experts at knowing what others know. This finding gives a clue that social networks can potentially be used to identify experts. Further evidence was seen in [21] where Campbell et al. showed that a graph based algorithm performed better than a simpler content based algorithm in finding experts. The graph used in this study was constructed from e-mail communications. Performances of the content and graph based algorithms were compared to human judgment of expertise.

Early works that used social networks to locate experts include Yenta [22]

and ReferralWeb [23]. The authors of Yenta propose a match making system with an underlying cluster building algorithm that uses referrals and users' "*similarity of interest*" [22] based on documents owned by individuals. Clusters of people are built relative to each individual. The cluster can be used to introduce oneself and pass messages to seek for expertise or help on a certain topic. ReferralWeb works in a similar way by building a social network based on sources such as the hyperlink structure of users' home-pages, co-authorship and citation lists of publications, newsgroup exchanges and organization charts. By utilizing this social network, ReferralWeb produces more focused search results for the seeker using referral chains. An example query illustrates this: "*what colleagues of mine, or colleagues of colleagues of mine, know about simulated annealing*" [23]? An interesting difference between Yenta and ReferralWeb is that the latter works with a global social network, while Yenta builds a cluster relative to each individual. Our work is similar to ReferralWeb in that it works with a global network graph.

In more recent works, graphs are created and various ranking algorithms are run on them to identify expert individuals [24, 25]. The authors of [24] construct an expertise graph based on e-mail communications and compare the performance of a few graph ranking algorithms including PageRank, HITS authority and a few simpler ones. Similarly, [25] compares the performance of PageRank, HITS and other simpler algorithms on a post-reply network constructed by parsing the discussion thread structure of the Java Forum. Both studies find that graph ranking algorithms can be used effectively to locate experts. We refer the interested reader to these works ([24, 25]) to review how PageRank and HITS are adopted to find expertise.

Our contribution is the social-expert algorithm which is specifically designed with the purpose of identifying experts in social networks. We suggest a future work of comparing social-expert's effectiveness against other ranking algorithms (such as PageRank and HITS) that are adopted for finding expertise in recent literature.

2.1 Graphs and expertise networks

We first review the specific graphs types on which ranking algorithms were executed in [24] and [25]. We then compare these to the graph model used to execute social-expert. The authors in [24] construct an expertise graph for a specific topic in two steps. First, all e-mails exchanged between two individuals on the topic are parsed and a relative expertise value is calculated between them. Next, an edge is created from the person with the higher expertise to the person with the lower expertise. The weight value is the magnitude of the relative expertise. These two steps are repeated for each node pair in the network. Adamic et al. in [25] construct a graph from the discussion thread structure of an Internet forum. The process involves creating an edge from the person that starts a discussion thread, to each person that participated in that discussion thread. Therefore, unlike [24], the edges flow from the person with the lower expertise (one who asks a question by starting a discussion thread), to those with higher expertise (those individuals that answer the question in the thread or participate in more discussion). We refer the interested reader to [26], which is a comprehensive study on the graph characteristics and expertise sharing aspects of a very large discussion thread network. We make an observation that in general graphs used for expertise identification are specific to a topic or domain. For example, an expertise graph for the "Java multi-threading" topic is likely to be very different from a graph for the topic: "Matlab".

We label the social-expert graph model as the "expertise network" model. Graphs of type expertise network are also topic-specific, but their edges do not necessarily flow from high to low expertise nodes, or from low to high expertise nodes. Each edge in an expertise network represents an explicit expertise evaluation through its weight and direction. For example, consider an edge from Alice to Bob with a weight of 8, where the weight value can lie between 0 and 10. This means Alice evaluated Bob to have 8 out of 10 expertise score. Let us assume that this edge existed in an expertise network for the topic of Java multi-threading. Then this edge may have been constructed when Bob advised Alice on a question on Java multi-threading and Alice evaluated Bob to have an expertise score of 8. In this work we have simulated some graphs representing such expertise networks.

We also obtained one graph from the Advogato community [3], which supports such explicit peer evaluations by members. In chapter 6 we propose a future work of developing a user interface that supports explicit peer evaluations through feedback. Our expertise network model is similar to a social network in that individuals are interconnected through some professional, academic or organizational work or correspondence. However the existence of edges and their corresponding weights represent peer evaluations. We now formalize our notion of a graph that represents an expertise network.

2.1.1 Formal definitions

We define a graph G as a set of nodes N and a set of edges E. In our study we use directed and weighted graphs, where each edge $e \in E$ contains a weight w and is directed from a source node src to a sink node $sink$. These are summarized below.

$$G = (N, E) \tag{2.1}$$

$$e \in E, e = (src, sink, w) \tag{2.2}$$

An *expertise network* is a special type of graph where each edge represents an explicit rating made by the source node (src) for the sink node ($sink$). The weight (w) on the edge (e) indicates the strength of the rating. Furthermore the expertise network is domain or topic specific. For example, in an expertise network on the topic *Matlab*, each edge represents the source node's opinion of the sink node's expertise on the *Matlab* programming language.

We use a ratings matrix R to define the edges received or produced by nodes. R is a square matrix with its rows and columns corresponding to nodes. An element $R[i, j] = \emptyset$ or $null$, if no edge exists between node i and j. An element $R[i, j] = w$, if an edge exists from node i to node j, $e = (i, j, w)$.

In an expertise network a person will typically rate some peers and also receive a set of ratings from his or her peers. We denote the set of peer nodes that rate

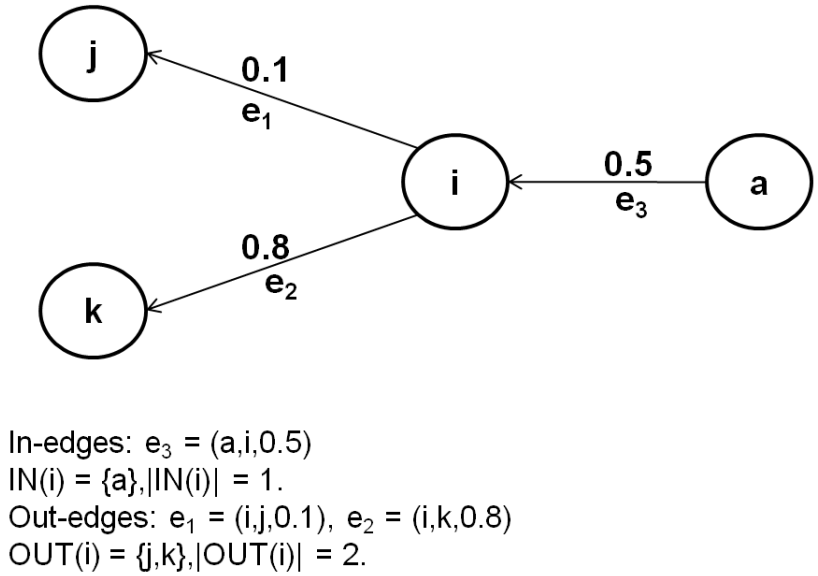

Figure 2.1: Inward and outward edges for Node i.

node i as $IN(i)$. The *in-degree*, or the number of incoming ratings (edges) for node i, is $|IN(i)|$. Similarly, $OUT(i)$ is the set of nodes that node i rates and $|OUT(i)|$ is the *out-degree*, or the total number of ratings made by i. Figure 2.1 clarifies these definitions.

Lastly, we place a restriction that each node i may have exactly one outgoing edge to another node j. However node i is then allowed to modify the weight of the rating or remove the rating altogether. Basically this means someone can make only one rating for another person's expertise (on a specific topic) and may choose to modify or delete that rating later.

CHAPTER 3
The Social-expert algorithm

The goal of social-expert is to use information from an expertise network and produce an absolute expertise rating for each person in the network. From the expertise network model of section 2.1, the information we have at hand are: the set of people in the network and explicit expertise ratings between various node pairs. We now describe how social-expert uses this information to calculate global expertise scores and explain our motivations at each step.

3.1 Peer-rating and Credibility

As a first attempt at producing an absolute expertise rating for a node, we take the simple average of the ratings the node has received. We denote this as the *peer-rating* component of social-expert. We use the notation $p(i)$ to denote the peer-rating of node i. Using the same graph notation from section 2.1.1 we can then describe this initial version of $p(i)$ as:

$$p(i) = \frac{\sum_{k \in IN(i)} R[k,i]}{|IN(i)|} \tag{3.1}$$

To improve on the simple average calculation of peer-rating we explored the intuition that some people in the network will be able to produce better expertise ratings. For example, some individuals in the network will have more experience and interactions with the community than others. Hence expertise ratings from such individuals are likely to be more accurate and should be weighted higher. We found evidence of this intuition in two expertise location studies [1, 27]. In a field study [1] at a Software company, McDonald and Ackerman identify so called "*Expertise Concierges*" who are "*key people who have very elaborated social networks*" [1]. Through their experience they can guide a person seeking expertise to the correct resource. Hence they are experts in knowing others' topic expertise. Similarly, [27] discusses a "*Contact Broker*" role: "*... brokers did not solve all problems themselves but were aware of activities and expertise of others and could direct clients to those*

experts" [27].

What we take away from this is that not everyone in the network is equally skilled at judging others' topic expertise. We label this as the *credibility* component of social-expert and use the notation $c(i)$ to denote the credibility of node i. We can now improve the simple average calculation of peer-rating in equation 3.1 to a weighted average based on credibility:

$$p(i) = \frac{\sum_{k \in IN(i)} (c(k) \times R[k,i])}{\sum_{k \in IN(i)} (c(k))} \tag{3.2}$$

In the above improved equation (3.2) we calculate a node's expertise by taking the weighted average of the ratings it received from peers. The weights imply that a rating received from a peer with high credibility will contribute more than a peer with low credibility.

Hence social-expert characterizes each node with two qualities: expertise on a certain topic measured through **peer-rating** and ability to rate or evaluate others accurately measured through **credibility**.

We make an observation here that our two component design is similar to the well known HITS Authority graph ranking algorithm for web search. In his work [5] describing HITS, Kleinberg distinguishes the quality of each web page with two components, a *hub* score and an *authority* score. The authority score indicates the extent to which a web page is a prominent source of quality content. The hub score indicates the extent to which the web page has links to good authorities. The authority and hub scores loosely translate to social-expert's peer-rating and credibility scores respectively.

The peer-rating component is the main output that we are seeking from running social-expert on a network. It reflects the expertise level of each node in the network. Note that the social-expert calculated peer-rating value for a node may not match the *true* expertise of that node. It is more accurate to say that peer-rating reflects the society's *opinion* of a node's expertise. Therefore, social-expert calculates expertise from the network's perspective. One can argue that over time, as a node receives more peer evaluations, its peer-rating should be a good estimate of true expertise. For example, in any active society, the members will accumulate more

interactions with other members over time, thereby making the society's estimation of any one member's expertise more accurate. To obtain the exact *true* expertise of someone it may be necessary to conduct examinations, personal interviews and so on.

3.2 Calculating credibility

Let us consider a simple case where node i has made only one expertise rating and it is to node j. The weight of this rating, r, and the credibility node i, $c(i)$, are on a scale of 0 to 10: $0 \leq r \leq 10$ and $0 \leq c(i) \leq 10$. Let us also assume that node j's true rating, $true(j)$, is 10. We now construct our credibility formula using some intuitions.

First we need to determine $true(j)$. As described in section 3.1 we cannot easily determine a node's *true* expertise. But we can substitute the peer-rating of node j, $p(j)$, which reflects the expertise of j from the network's perspective. Therefore, $true(j) = p(j)$. An implication of this substitution is that as well as peer-rating, credibility will also be calculated from the network's perspective.

Intuitively, the credibility of node i, $c(i)$, should be high if r is close to $p(j)$ and low if r is away from $p(j)$. Furthermore $c(i)$ should be 10 (maximum) if $r = p(j)$ and $c(i)$ should be 0 (minimum) if r is the farthest possible distance away from $p(j)$, i.e. $r = 0$. Therefore we know that $c(i)$ is a function of the difference between $p(j)$ and r. These intuitions are summarized below:

$$\begin{aligned} c(i) &= f(|p(j) - r|) \\ c(i)_{max} &= f(0) \quad = 10 \\ c(i)_{min} &= f(10) = 0 \end{aligned}$$

We still need to determine the value of $c(i)$ when $(|p(j) - r|)$ is > 0 and < 10. A very simple function would be if $c(i)$ reduces linearly as the difference between $p(j)$ and r increases. This relationship is shown in equation 3.3.

$$c(i)_r = -(|p(j) - r|) + 10 \quad (0 \leq r \leq 10\,,\, 0 \leq p(j) \leq 10) \tag{3.3}$$

The above equation presents the following problem. Let us assume that 100 nodes rate j on a scale of 0 to 10 and that $true(j) = 8$. Then it is unlikely that all 100 nodes will rate j exactly as 8 since there will naturally be some variations in node j's evaluation. We observe that this quantitative rating is really a translation of the opinion of node j's evaluator. In real life this opinion is likely to be largely qualitative, formed from past interactions between the evaluator and the person being rated. When converting this qualitative experience to a number between 0 and 10, it is highly likely that there are some variations. We have listed some reasons below based on our literature survey:

Performance from past experience: let us assume node j was able to solve a problem in the past for node i. Hence node i had a positive experience and rated node j as 9. But perhaps node j could not answer a question presented by node k and therefore node k rated node j as 6. The study in [1] shows performance as an important factor in expert selection once candidates have been identified.

Evaluator's own expertise level: the evaluator's own expertise may influence his or her rating for another person. For example, Let us assume node e herself is very experienced on the subject matter and has high expertise. She may think of node j as average and give a rating of 6. Perhaps another node, n, who is a novice thought highly of j from a past experience and gave a rating of 10. Or, it could be that node j was not particularly patient or helpful with another novice node m who asked a naive question. Consequently, it caused m to rate j as 5. Paepcke points out in his study [27] that for novice seekers, experts were matched carefully to those "*with an interest in teaching or at least some tolerance for naive questions*" [27].

Bias: variation in ratings may also surface from bias evaluations. For example an individual may have a tendency to consistently give others higher than true ratings.

Because $p(j)$ is a weighted average of all the individual ratings that j receives, the above variations will affect the final value of $p(j)$. Ideally we do not want these variations to affect credibility too much. As long as the rating is close to 8 ($p(j)$), we want to consider it an accurate rating and deem the rater as credible. Only if the rating is very much away from 8 ($p(j)$) we want to consider the rater less credible.

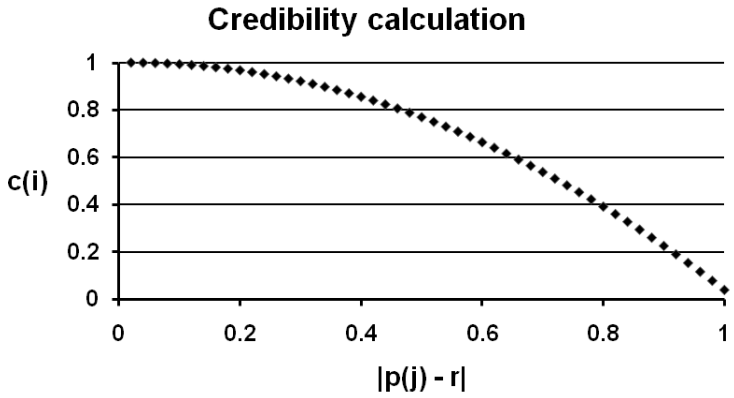

Figure 3.1: Credibility ($c(i)$) as a function of difference ($|p(j) - r|$).

This intuition is not captured in our linear credibility calculation in equation 3.3. We observe that a quadratic relationship between credibility $c(i)$ and $|p(j) - r|$ does describe the desired intuition. Equation 3.4 describes this relationship.

$$c(i)_r = 1 - (p(j) - r)^2 \quad (0 \leq r \leq 1\,,\, 0 \leq p(j) \leq 1) \tag{3.4}$$

Note that we have made another change in equation 3.4; both the rating weight r and the peer-rating $p(j)$ are now on a continuous scale between 0 to 1. This normalization allows for better adoption of the social-expert algorithm into various rating scales. Figure 3.1 shows that with a quadratic relationship, $c(i)$ stays relatively flat when the difference is close to 0. For example, even when the difference is 0.3, the rater's credibility ($c(i)$) stays over 0.9 ($1 - (0.3)^2$). Once the difference is ≥ 0.5, $c(i)$ starts to decrease rapidly.

So far we have built the credibility formula for node i assuming that he or she has made only one rating. In the general case where a node may make multiple ratings, we will need to average the credibility over all the outgoing ratings. We express this below in equation 3.5:

$$c(i) = \frac{\sum_{j \in OUT(i)} (1 - (R[i,j] - p(j))^2)}{|OUT(i)|} \tag{3.5}$$

For convenience the peer-rating equation (3.2) derived in section 3.1 is repeated below as equation 3.6:

$$p(i) = \frac{\sum_{k \in IN(i)} (c(k) \times R[k,i])}{\sum_{k \in IN(i)} (c(k))} \tag{3.6}$$

The two components of social-expert, credibility and peer-rating, are described by equations 3.5 and 3.6 respectively. In both equations the rating weight, $R[i,j]$, lies between 0 and 1: $0 \leq R[i,j] \leq 1$. This implies that the calculated peer-rating and credibility for any node n, $p(n)$ and $c(n)$, will also lie between 0 and 1. Therefore, for all $n \in N$, $0 \leq p(n) \leq 1$ and $0 \leq c(n) \leq 1$.

3.3 Execution

Social-expert is an iterative algorithm. Its input is an expertise network graph $G = (N, E)$ in the form of a ratings matrix R. In each iteration social-expert calculates the peer-rating $p(i)$ and credibility $c(i)$ score of every node $i \in N$. As the number of iterations increases we expect the scores to converge and stabilize to *final* values, after which there are no more changes to any node's scores. The output of social-expert is a vector SR where $SR(i) = p(i), c(i)$ for all nodes $i \in N$. Each value pair $p(i), c(i)$ in SR is the *final* peer-rating and credibility score for node i. The pseudo-code for social-expert is shown below. Note that the peer-rating $p(i)$ will be *null* if node i has 0 in-degree (i.e. it received 0 peer evaluations). Similarly, the credibility $c(i)$ will be *null* if the node has 0 out-degree (i.e. it did not evaluate any peers). Therefore, in our implementation of social-expert we first check for these conditions before proceeding to calculate peer-rating or credibility.

Input: A set of nodes N. Ratings matrix R. Credibility vector c, Peer-rating vector p.
Output: SR: a $n \times 2$ vector of nodes' $p(i)$ and $c(i)$.

while *AllScoresNotStable()* **do**
  foreach *node* $i \in N$ **do**
    // Peer-rating: get nodes that have evaluated i:
    $IN = getINNodes(i)$;
    // Variables to store weighted sum and total weights:
    $WSum = 0$;
    $weights = 0$;
    foreach *node* $k \in IN$ **do**
      // Calculate weighted sum ($\sum_{k \in IN(i)}(c(k) \times R[k,i])$):
      $WSum = WSum + c[k] \times R[k,i]$;
      $weights = weights + c[k]$;
    end
    // Calculate peer-rating score using equation 3.6:
    $p[i] = WSum/weights$;
    // Credibility: get nodes i has evaluated:
    $OUT = getOUTNodes(i)$;
    $outDegree = size(OUT)$;
    // Variable to store sum of difference:
    $DSum = 0$;
    foreach *node* $j \in OUT$ **do**
      // Calculate sum of difference
      // ($\sum_{j \in OUT(i)}(1 - (R[i,j] - p(j))^2)$):
      $diff = (R[i,j] - p[j])$;
      $Dsum = Dsum + (1 - (diff \times diff))$;
    end
    // Calculate credibility score using equation 3.5:
    $c[i] = DSum/outDegree$;
    // Record peer-rating and credibility scores in SR:
    $SR[i,0] = p[i]$;
    $SR[i,1] = c[i]$;
  end
end

Algorithm 1: Social-expert pseudo-code.

CHAPTER 4
Graphs and Measurements

The goal of the social-expert algorithm is to determine global expertise and credibility levels of users in an expertise network. In this chapter we discuss our experiments for measuring the effectiveness of social-expert in achieving this goal. We start by discussing the input datasets used for testing, followed by data collection and analysis techniques.

4.1 Datasets for testing Social-expert

Ideally we would start an online community which supports expertise rating among members and use data from it to test social-expert. Unfortunately this is impractical as it would take too long to gain a large number of users. Instead we have tested social-expert on graphs constructed from simulations and real world datasets. In the subsequent sections we refine our graph model further, then describe and motivate our input dataset choices.

4.1.1 Building Graphs

To describe how we have built our input graphs we differentiate between two components of a graph: *topology* and *weight*. The topology component represents the structure of the graph and consists of a set of nodes and *unweighted* directed edges. The weight component represents the weight values to be assigned to all edges. A weight value indicates the magnitude of the relation represented by the edge. Since we are working with expertise networks, the relation expressed by an edge is an explicit expertise rating from one person to another. Hence the weight value indicates the rater's opinion of the expertise level of the person being rated (on a given topic). We can now formally express a graph G in terms of these components:

$$G = (topology, weight)$$

Recall from section 2.1.1 that we defined a graph G as a set of nodes N and directed weighted edges E: $G = (N, E)$. Also recall that an edge $e \in E$ consists of a source node (src), a sink node ($sink$) and a weight (w) indicating the magnitude of the relation: $e = (src, sink, w)$. We will now use these to construct a graph. We start building the graph's *topology* component by obtaining a set of nodes N and a set of *unweighted* directed edges U. An unweighted edge $u \in U$ consists of just a source and a sink node, $u = (src, sink)$.

We then build the *weight* component by obtaining a set W containing weight values (w) and unweighted edge (u) pairs. Each element $x \in W$ consists of the weight value w to be assigned to the edge u: $x = (u, w)$. By assigning the weight value w to its corresponding unweighted edge u, we construct the weighted directed edge $e = (src, sink, w)$. We then add this weighted directed edge to the set E. By performing this weight assignment for each element $x \in W$ we obtain the full set of weighted directed edges E.

We can now combine the set of nodes N from the graph's *topology* and the set of weighted edges E from the graph's *weight* to form the final graph as defined in section 2.1.1:

$$G = (N, E)$$

We provide a quick example of building a graph using the above process. We construct the *topology* component from a set of nodes N and unweighted edges U.

$$N = \{\mathbf{A}lice, \mathbf{B}ob, \mathbf{C}arol, \mathbf{D}ave\} = \{A, B, C, D\}$$

$$U = \{(A, B), (B, A), (A, C), (B, C), (B, D), (C, D)\}$$

The above sets can be simulated or obtained from existing online graph datasets. We construct the *weight* component by attaching a weight value w to each unweighted edge $u \in U$. The weight value (w) can be obtained from simulation, or from real existing online expertise networks that support explicit peer evaluations. We store w and u as pairs in the set W:

$$W = \{((A, B), 0.6), ((B, A), 0.9), ((A, C), 0.5),$$

Graphs				
Properties	G_{SMALL}	$G_{FACEBOOK}$	$G_{ADV-SIM}$	$G_{ADVOGATO}$
topology data	Simulated	Facebook NO [28]	Advogato [29]	Advogato [29]
weight data	Simulated	Simulated	Simulated	Advogato [29]
Number of nodes	40	63731	7430	7430
Number of edges	1249	1545686	57201	57201
Average degree	62.5	48.5	15.4	15.4
Maximum degree	70	2113	977	977
Average link symmetry	0.79	0.90	0.59	0.59
Edge weight data type (w)	Continuous Values $0 \leq w \leq 1$			Discrete Values $w = \{1.0(Master), 0.66(Journeyer), 0.33(Apprentice), (Observer)\}$

Table 4.1: Input graphs for testing social-expert.

$$((B, C), 0.75), ((B, D), 0.2), ((C, D), 0.1)\}$$
$$(u = (src, sink),\ x \in W,\ x = (u, w))$$

From each pair $x \in W$ we obtain a weighted edge e. We add e to the set E to obtain our graph definition $G = (N, E)$:

$$N = \{A, B, C, D\}$$
$$E = \{(A, B, 0.6), (B, A, 0.9), (A, C, 0.5),$$
$$(B, C, 0.75), (B, D, 0.2), (C, D, 0.1)\}$$
$$(e \in E,\ e = (src, sink, w))$$

Using the techniques outlined above we constructed 4 input graphs: G_{SMALL}, $G_{FACEBOOK}$, $G_{ADV-SIM}$ and $G_{ADVOGATO}$. To ensure comprehensive testing our graphs and their underlying datasets vary in terms of size, topology and weight components. We chose small, medium and large graphs with about 40, 7000 and

64000 nodes respectively. Table 4.1 shows various properties of these graphs, including the source data for weight and topology. We now describe these datasets and the corresponding graphs in more detail.

4.1.2 Input graphs for testing Social-expert

G_{SMALL} is small graph where we simulated 40 randomly interconnected nodes. Both the topology and weight components of this graph were simulated. $G_{FACEBOOK}$ is a much larger graph that we constructed from a dataset which was made available to the research community by Mislove et al. in [28]. This dataset was collected from a popular social networking site called Facebook [30]. Mislove et al. used the Facebook New Orleans regional network to study inter user activity patterns. The dataset contains two components: friendship links among users in the network and the number of wall posts[1] between mutual friends. The latter represents inter user activity which was being studied by the authors. We used the first component of this dataset to construct the topology of $G_{FACEBOOK}$. Because there are no expertise ratings among users in Facebook, we simulated the weight component for this graph.

Graphs $G_{ADVOGATO}$ and $G_{ADV-SIM}$ were constructed from a publicly available graph dataset [29] that represents the Advogato online community [3]. Because this community supports explicit expertise ratings among members, we obtained both weight and topology for $G_{ADVOGATO}$ directly from this dataset. We also chose to build another graph $G_{ADV-SIM}$ where we only used the Advogato dataset for topology and simulated the weight component. The weight component was simulated in $G_{ADV-SIM}$ for data analysis purposes (described further in section 4.2.3). The Advogato community was started by Raph Levien for software developers who contribute to free software. In his original (unfinished) PhD Thesis [13], Levien discusses the implementation of his trust metrics research on the Advogato online community. Advogato has since been discussed in a series of research works in the area of trust metrics and social networks. Examples include the widely cited work [9] of Golbeck on inferring trust from network structure and Ziegler and Lausen's

[1] Wall posts: a form of peer messaging on Facebook.

trust algorithm called Appleseed [32].

4.1.3 Topology of input graphs

We constructed the topology of G_{SMALL} by simulating outgoing edges from each node with a *uniform* degree distribution. Therefore each node contained approximately the same degree (number of outgoing and incoming edges). The interconnections among nodes were chosen randomly. The degree was determined using a *density* parameter of 0.8; more specifically, each node will have outgoing edges to about 80% of the network. With a graph size of 40 nodes, this meant each node in G_{SMALL} contained about 64 edges total: 32 incoming and 32 outgoing edges. Compared to $G_{FACEBOOK}$ and $G_{ADVOGATO}$, having 64 edges per node is relatively high. We chose such high density since an individual in a small network is likely to know most other individuals in that network. In contrast, the nodes with the highest degrees in $G_{FACEBOOK}$ and $G_{ADVOGATO}$ are only directly connected to 3% and 13% of the remaining entire network (please see table 4.1 for details).

A series of research works [33, 34, 35, 36] have shown that online network topologies exhibit certain special characteristics. They also show that naturally occurring online social networks do *not* exhibit a uniform degree distribution like G_{SMALL}. Hence, to make our tests more realistic we obtained the topology of $G_{FACEBOOK}$, $G_{ADVOGATO}$ and $G_{ADV-SIM}$ from real world graph datasets. We summarize the special characteristics of naturally occurring networks below.

Power-law degree distribution : the probability P that a node has degree k (the number of incoming and outgoing edges from the node) is determined by a power law $P(k) = Ak^{-\gamma}$, where γ is the power-law coefficient and is greater than 1. Furthermore Li et al. in [37] propose a special type of power-law network with scale-free properties, which asserts that nodes with high degrees tend to be connected to other nodes with high degrees.

Small-world: the diameter of a small-world network (average number of hops to reach a node from any other node) is very small. In his famous experiment Milgram originally demonstrated that the average number of hops between two Americans is 6 [33]. Kleinberg models this phenomenon from an algorithmic perspective

to explore navigability in such networks [38].

High clustering: another characteristic of a small-world network is that they have tightly connected clusters and a densely connected core. The core is a group of well connected nodes with high degrees. If the core nodes are removed from the network, it would result in the network breaking into many isolated enclaves. The clustering-coefficient provides a measure of the level of clustering in a network. Watts and Strogatz demonstrate a social network model [34] that exhibits the small-diameter and high-clustering properties.

Adamic et al's work [36] on Club Nexus, an online student community of Stanford University, demonstrates the small-world, power-law and high-clustering properties. It is interesting to note the authors' comment:

> *"In the case of Club Nexus the clustering coefficient of 0.17 is 40 times higher than it would be for a random network with the same number of users and connections"* [36].

Mislove et al. in their comprehensive 2007 work [35] examined topologies of 4 popular online social networks (Flickr , YouTube , LiveJournal and Orkut) and confirmed the power-law, small-world, scale-free and high clustering properties in a total of 11.3 million users and 328 million links. Our $G_{FACEBOOK}$ dataset is obtained from the subsequent work [28] of Mislove et al. that studied user interactions in Facebook.

Based on this thread of research work we can be adequately confident that the topology component of our input datasets are representative of the desirable characteristics that would for example, be exhibited by an online expertise network where the social-expert algorithm can be deployed.

4.1.4 Adding expertise ratings to graphs

After constructing the graph topology we are left with the task of augmenting a weight value to each edge. Recall that the weight value on an edge represents an expertise rating made from the source node to the sink node. As shown in table 4.1 we have simulated these weight values for graphs G_{SMALL}, $G_{FACEBOOK}$ and

$G_{ADV-SIM}$. For $G_{ADVOGATO}$, due to the nature of the underlying dataset, this data was available on hand. We now discuss the augmentation of ratings further.

4.1.4.1 G_{SMALL}, $G_{FACEBOOK}$ and $G_{ADV-SIM}$ - simulating expertise ratings

To simulate expertise ratings we first produce a target expertise value for each node (person). This represents the true expertise level for that person and should be close to the peer-rating value once social-expert is run on the graph. For a node i, we denote his or her simulated true expertise value as $p_{TRUE}(i)$. In section 2.1.1 we have defined this weight value to lie between 0 and 1 in a continuous scale. We have distributed $p_{TRUE}(i)$ with a Gaussian distribution with mean $(\bar{x}) = 0.5$ and standard deviation $(\mu) = 0.25$. Figure 4.1 shows this distribution for graph $G_{FACEBOOK\,0.9}$[2].

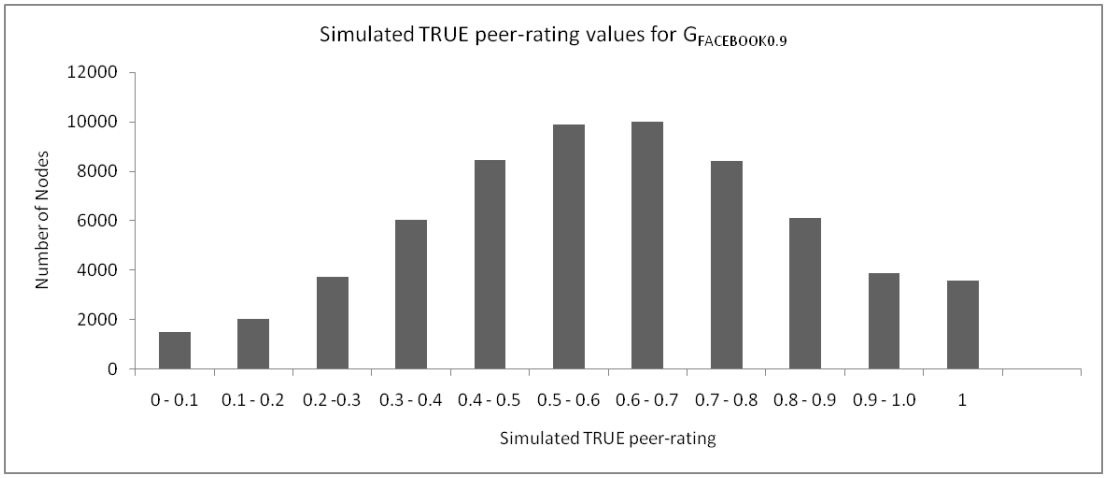

Figure 4.1: Simulated p_{TRUE} values for $G_{FACEBOOK\,0.9}$.

In chapter 3 we described each node with two qualities: its expertise level (peer-rating) and its credibility. By having distinct scores for these qualities, we are able to express them independently of each other. For example, a person may be an expert on a certain topic but a poor judge of others' expertise. Similarly a person may be an expert in knowing who knows what in the network, regardless of how well he or she knows a certain topic. Therefore as the next step of simulating expertise ratings, we label each node randomly as having *good* or *bad* credibility independent of its true expertise level. Similar to $p_{TRUE}(i)$, we now assign a true

[2] $G_{FACEBOOK\,0.9}$ is a version of $G_{FACEBOOK}$, please see table 4.2 for more details.

credibility value $c_{TRUE}(i)$ for each node (person) with the following rules. A *good* node will have a true credibility value randomly selected between 0.95 and 1.0. A *bad* node will have a true credibility value randomly selected between 0 and 0.3. These rules are expressed below:

$$\text{If } i \text{ is } bad \rightarrow 0 \leq c_{TRUE}(i) \leq 0.3$$
$$\text{If } i \text{ is } good \rightarrow 0.95 \leq c_{TRUE}(i) \leq 1.0$$

Our intuition is that a *good* node should produce more *credible* ratings that are close to $p_{TRUE}(j)$ (where j is the node being rated). Similarly a *bad* node should produce more *non–credible* ratings that are away from $p_{TRUE}(j)$. The exact ratio of *credible* to *non–credible* ratings is the true credibility we assigned: $c_{TRUE}(i)$ (where the node performing the rating is i). For example, if node i is a *good* node and has $c_{TRUE}(i) = 0.98$, then 98% of its outgoing edges will be *credible*. Similarly if node i is a *bad* node and has $c_{TRUE}(i) = 0.2$, then only 20% of its outgoing edges will be *credible* and the rest will be *non–credible*. We formally define *credible* and *non–credible* ratings further in sections 4.1.4.2 and 4.1.4.3.

Recall that our graph definition is $G = (N, E)$. After generating $p_{TRUE}(i)$ and $c_{TRUE}(i)$ for all nodes $i \in N$ and labeling each edge $e \in E$ as *credible* or *non–credible*, we are ready to augment each edge with a weight value (w). The subsequent sections describe how the exact weight values are chosen.

4.1.4.2 Credible ratings

A *credible* rating represents a node (person) making a good judgment of another node's true expertise level. If the node being rated is i, then the weight value (w) for a *credible* rating is chosen with a uniform random distribution of $p_{TRUE}(i) \pm 0.05$. Since we limit the rating weight value (w) between 0 and 1, the actual value of w for a *credible* rating is chosen in the range: $max(0,\ p_{TRUE}(i) - 0.05) \leq w \leq min(1,\ p_{TRUE}(i) + 0.05)$.

4.1.4.3 Non-credible ratings

A *non–credible* rating represents a node (person) making a bad judgment of another node's true expertise level. Therefore, we want the weight value (w) for

a *non–credible* rating to be away from the true expertise level ($p_{TRUE}(i)$) of the node being rated. In addition, the lower the credibility of the source node (the node making the rating), the further away the weight value (w) of the rating should be from $p_{TRUE}(i)$.

More formally, when a node i is being rated we want the following two conditions to be true. First, in general the difference $|w - p_{TRUE}(i)|$ for a *non–credible* rating should be larger than the difference for a *credible* rating. Second, if node y has lower credibility than node x, then the difference $|w - p_{TRUE}(i)|$ for a rating made by node y should by larger than the difference for a rating made by node x. These two conditions are described below in expressions 4.1 and 4.2.

$$|w_{NON-CREDIBLE} - p_{TRUE}(i)| > |w_{CREDIBLE} - p_{TRUE}(i)| \tag{4.1}$$

$$c_{TRUE}(y) < c_{TRUE}(x) \rightarrow |w(y) - p_{TRUE}(i)| > |w(x) - p_{TRUE}(i)| \tag{4.2}$$

Let us assume that node x is rating node i. Then we can satisfy the above conditions by randomly choosing the *non–credible* edge weight value (w) from a Gaussian distribution with mean $p_{TRUE}(i) + 5$ and standard deviation $c_{TRUE}(x)$. For example, if $p_{TRUE}(i) = 0.7$ and $c_{TRUE}(x) = 0.25$, then we will choose a weight value randomly from a Gaussian distribution with $\bar{w} = 5.7$ and $\mu = 0.25$. This satisfies the first condition since in general a value from this distribution will be close to the mean of 5.7. It is unlikely for this value to be in the range of a *credible* edge: 0.7 ± 0.05.

To show that this strategy also satisfies second condition we present the following example. Let us assume node y makes a rating to node i and that node y has a true credibility $c_{TRUE}(y) = 0.1$. Then we will choose a weight value randomly from a Gaussian distribution with $\bar{w} = 5.7$ and $\mu = 0.1$. Because the standard deviation is lower for node y (0.1) than node x (0.25), the weight value (w) we pick for y will have a greater probability to be further away from $p_{TRUE}(i)$.

To obtain the final weight value (w) we calculate the absolute difference between the mean value and the random weight value obtained from the Gaussian distribution described above. We then normalize this difference over a scale of 0 to

1 by dividing by 5. For example, if the weight value chosen for x was 5.3, then the difference from the mean value is: $|5.7 - 5.3| = 0.4$. We normalize it to a scale of 0 to 1 by dividing by 5: $\frac{0.4}{5} = 0.08$. As another example, if the weight value chosen for y was 5.8, then the difference from the mean value is: $|5.7 - 5.8| = 0.1$. As before, we normalize it to a scale of 0 to 1 by dividing by 5: $\frac{0.1}{5} = 0.002$. Finally, we must take into account that the rating weight value (w) is between 0 and 1. Therefore we choose the actual value of w for a *non–credible* rating as: $min(0, max(1, w))$.

Conditions 4.1 and 4.2 imply that having more *good* nodes in a graph will result in having more *credible* edges. In our simulations we vary the ratio of *good* and *bad* nodes from 100% *good* nodes to 50% *good* nodes resulting in different amounts of *credible* and *non–credible* edges. Table 4.2 shows different versions of graphs G_{SMALL}, $G_{FACEBOOK}$ and $G_{ADV-SIM}$, which vary in the % of good nodes and *credible* edges. We named each graph in the format: $G_{graphname\ <distribution>}$, where $_{<distribution>}$ indicates the ratio of *good* nodes in the graph.

4.1.4.4 Expertise ratings for $G_{ADVOGATO}$

The method of simulating expertise ratings discussed in the previous section was not applied to graph $G_{ADVOGATO}$. In this section we describe how ratings were augmented to $G_{ADVOGATO}$. We start with some background of the Advogato community [3].

Advogato is interesting from a research perspective especially for its tiered certification or rating system. Members of the Advogato community can certify any other member at levels *observer*, *apprentice*, *journeyer* or *master*. We observe from the definitions of these levels [39] that they map well to the talent, skill and experience components of our expertise definition from chapter 1. An excerpt from [39] showing these definitions is provided in appendix A. This explicit peer certification system also maps well to the weight component of our expertise graph. We can convert these certification levels directly to the edge weights representing expertise ratings among nodes. For a node that certifies another node as master, we create an edge with weight $w = 1.0$. Similarly, we map a journeyer certification to $w = 0.66$, apprentice to $w = 0.33$ and observer to $w = 0$.

Graph	Number of *good* nodes ($0.95 \leq c_{TRUE}(i) \leq 1.0$)	Number of *credible* edges
G_{SMALL}		
$G_{SMALL\,1.0}$	40 (100%)	1230 (98.5%)
$G_{SMALL\,0.9}$	34 (85.0%)	1055 (84.5%)
$G_{SMALL\,0.8}$	31 (77.5%)	996 (79.7%)
$G_{SMALL\,0.7}$	27 (67.5%)	901 (72.1%)
$G_{SMALL\,0.6}$	24 (60.0%)	779 (62.4%)
$G_{SMALL\,0.5}$	19 (47.5%)	679 (54.4%)
$G_{FACEBOOK}$		
$G_{FACEBOOK\,1.0}$	63731 (100%)	1523307 (99.1%)
$G_{FACEBOOK\,0.9}$	57345 (90.0%)	1390625 (90.0%)
$G_{FACEBOOK\,0.8}$	50968 (80.0%)	1261370 (81.6%)
$G_{FACEBOOK\,0.7}$	44541 (70.0%)	1122570 (72.6%)
$G_{FACEBOOK\,0.6}$	38585 (60.5%)	1009401 (65.3%)
$G_{FACEBOOK\,0.5}$	31439 (49.3%)	858771 (55.6%)
$G_{ADV-SIM}$		
$G_{ADV-SIM\,1.0}$	7430 (100%)	56624 (99.0%)
$G_{ADV-SIM\,0.9}$	6704 (90.2%)	51348 (89.8%)
$G_{ADV-SIM\,0.8}$	5897 (79.4%)	46053 (80.5%)
$G_{ADV-SIM\,0.7}$	5255 (70.7%)	41095 (71.8%)
$G_{ADV-SIM\,0.6}$	4484 (60.4%)	34804 (60.8%)
$G_{ADV-SIM\,0.5}$	3751 (50.5%)	31436 (55.0%)

Table 4.2: Simulated graph versions.

4.1.4.5 Advogato global certifications

Similar to our effort, a key feature of the Advogato network is that each member receives a global certification at one of four levels. We provide a brief overview of this process in the interest of comparison with social-expert.

The Advogato certifications are calculated using a trust metric algorithm [40, 13] proposed by Raph Levien. The algorithm is run once for each level: master, journeyer, apprentice and observer. At each level the community is mapped to a graph with the members as nodes and peer certifications as directed edges. Only edges at the current level or higher are considered. For example, master global certifications are calculated with master peer certifications only, while all peer certifications are considered when calculating observer global certifications.

Levien then assigns capacity values to a small set of highly trusted seed nodes.

The capacity value loosely translates to the amount of trust that can be placed on each node. This capacity is then propagated via the Ford-Fulkeson network flow algorithm, where the source is the seed node set and the sink is a virtual super-sink node. All nodes between the source and sink receive capacity (trust) values which diminish as their distance from the seed increase and connectivity to the seed decrease. The final global certification for each node is assigned based on the capacity value of the node.

Privileges in the Advogato community are based on a node's global certification. For example, nodes certified as observer have read only access to the community's discussion threads. Also, nodes certified as apprentice can only post comments on the community discussion board, while master and journeyer nodes can post both comments and stories. We differentiate Levien's trust metric algorithm by observing that it was designed for attack resiliency against malicious nodes [13, 39], not necessarily for global expertise ratings. We compare trust metrics research to social-expert further in chapter 6.

Other works that have used the Advogato dataset include a comprehensive evaluation [41] of four different trust metrics (including Ebay, Advogato, Moletrust and Pagerank). Another work [8] used the Advogato dataset to test their new model of propagating trust in mobile devices.

4.2 Measuring Social-expert performance

Having described the input datasets we now proceed to discuss the tests we plan to run on them. The goal of social-expert is to assign global expertise values to users in an expertise network. To determine the effectiveness of social-expert in achieving this goal we seek answers to the following two questions. *Output Precision*: are the final peer-rating and credibility output for each node convergent to a stable value? Furthermore, are the output values identical regardless of the initial peer-rating and credibility estimates? *Output Accuracy*: do the expertise and credibility output for a node reflect the true expertise and credibility of that node?

4.2.1 Output precision

Recall from chapter 3 that the output of social-expert is a vector SR where the i-th coordinate of the vector: $SR(i)$ is the value pair $(p(i), c(i))$, for all nodes $i \in N$. The values $p(i)$ and $c(i)$ represent the peer-rating and credibility of node i. In order for $p(i)$ and $c(i)$ to be meaningful we must ensure that as social-expert is run on the network over many iterations, both $p(i)$ and $c(i)$ converge to stable values (for all $i \in N$). In addition, we are interested in measuring whether the output values are consistent regardless of the initial peer-rating and credibility values used on the first iteration.

4.2.1.1 Initial and final values

To calculate the output at iteration n, social-expert will rely on the values calculated at iteration $n-1$. This implies that before we can even run the very first iteration ($n = 1$), we must establish an initial ($n = 0$) peer-rating and credibility value for each node in that graph.

To formalize these notions, let us denote the peer-rating and credibility of node i at iteration n as $p(i)_n$ and $c(i)_n$ respectively. To calculate $p(i)_n$ and $c(i)_n$ we must know the values of $p(j)_{n-1}$ and $c(j)_{n-1}$, for every node j that is a neighbor of node i. This means, to calculate $p(i)_1$ and $c(i)_1$, we need to know the values of $p(j)_0$ and $c(j)_0$. As $n = 1$ is the very first iteration, we use $p(j)_0$ and $c(j)_0$ to denote the *initial* peer-rating and credibility for every node $j \in N$. Finally after many iterations of social-expert has been run, we consider the output as *final* and denote it as $p(j)_\infty$ and $c(j)_\infty$ (for every node $j \in N$)

Unfortunately, $p(j)_0$ and $c(j)_0$ is not readily at hand and needs to be estimated. To estimate the initial credibility ($c(j)_0$) for a node j one strategy is to allow the node benefit of the doubt and render it as fully credible, i.e. $c(j)_0 = 1.0$. Another option is to allow initial credibility to be a value between 0 and 1, such as 0.5. A reasonable initial expertise rating (peer-rating) $p(j)_0$ estimate should be the simple average of the expertise ratings made by neighbors of node j. The Advogato network (discussed in section 4.1.4.4) initializes new nodes to initial trust of 0.

An estimation method depends on the following key question: for a given

graph $G = (N, E)$, do $p(j)_\infty$ and $c(j)_\infty$ for all nodes $j \in N$ depend on $p(j)_0$ and $c(j)_0$ (for all nodes $j \in N$)? For example, let us assume that we perform a test, *test* 1, where we initialize all nodes to have peer-rating and credibility of 0.5, $p(j)_0 = 0.5, c(j)_0 = 0.5$ for all $j \in N$. Then we perform another test, *test* 2, for the same graph where we initialize all nodes to have peer-rating and credibility of 0.95, $p(j)_0 = 0.95, c(j)_0 = 0.95$ for all $j \in N$. Then the question we ask is: will $p(j)_\infty$ and $c(j)_\infty$ for *test* 1 and *test* 2 be different or same for each node $j \in N$? If they are different then we must estimate $p(j)_0$ and $c(j)_0$ carefully to ensure we receive the most accurate results. However, if $p(j)_\infty$ and $c(j)_\infty$ for *test* 1 and *test* 2 are the same then it does not matter what values we assign to $p(j)_0$ and $c(j)_0$ (since the final result is identical).

Less formally, we note that the choice of careful initial value selection is irrelevant if the final outputs for each node after many iterations always converge to the same values. This will be the ideal case as it means social-expert output is only dependent on the graph topology and weights; not on the initial peer-rating and credibility values that each node is initialized to.

4.2.1.2 Measuring output precision

Based on the above discussion we are interested in measuring the following two aspects of output precision. *Convergence*: as we execute social-expert on a graph for many iterations whether the output peer-rating and credibility converge to stable values for each node. More formally, for a graph $G = (N, E)$, we want to test if $p(j)_\infty$ and $c(j)_\infty$ for all nodes $j \in N$ converge to stable values. *Convergence to the same value*: whether the output peer-rating and credibility values after many iterations depend on initial values. More formally, we also want to test if $p(j)_\infty$ and $c(j)_\infty$ for all nodes $j \in N$ vary, if we vary $p(j)_0$ and $c(j)_0$.

To test for *convergence* at the nth iteration we calculate a running average up to $n-1$ iterations. We denote the peer-rating average at the $n-1$ iteration for node j as $\overline{p(j)_{n-1}}$ and the credibility average as $\overline{c(j)_{n-1}}$. We then calculate the difference between these averages and the social-expert output value at the nth iteration:

$$p(j)_{DIFF,n} = |\overline{p(j)_{n-1}} - p(j)_n|$$

$$c(j)_{DIFF,n} = |\overline{c(j)_{n-1}} - c(j)_n|$$

We can say that the peer-rating and credibility values have converged for node j if the differences, $p(j)_{DIFF,n}$ and $c(j)_{DIFF,n}$, approach 0 as the number of iterations (n) approaches ∞:

$$\text{Convergence for } p(j) : lim_{n\to\infty}\, p(j)_{DIFF,n} = 0.$$
$$\text{Convergence for } c(j) : lim_{n\to\infty}\, c(j)_{DIFF,n} = 0.$$

Less formally, as the number of iterations increase, the difference between the value at nth iteration and the cumulative average up to $n-1$ iterations should approach 0. In our implementation of the convergence test we do allow for some fluctuations of $p(j)_{DIFF,n}$ and $c(j)_{DIFF,n}$. As long as they *eventually* reach and stay near 0 we say $p(j)$ and $c(j)$ has converged.

To test for *convergence to the same value* we perform a series of tests on each input graph dataset. For each test we only vary the initial peer-rating and credibility value, $p(j)_0$ and $c(j)_0$, for all nodes $j \in N$. After running social-expert for 30 iterations we check whether the output values $p(j)_{30}$ and $c(j)_{30}$ are identical across all tests for each node $j \in N$. If so, we can conclude that social-expert outputs indeed converge to the same value regardless of how the nodes are initialized. To ensure that $p(j)_{30}$ and $c(j)_{30}$ are identical across all tests we can use the *t-test* to check for statistically significant differences between any two tests. If our p-value (α) from the t-test is larger than 0.95, then we can assert the null hypothesis that there are no differences in the final outcome between any two tests. Table 4.3 shows the combinations of initial values we plan to test with. For test 7 we initialize each node to a random value between 0 and 1. For test 8 we initialize the peer-rating of each node to the average expertise ratings it received.

4.2.2 Output accuracy

Our contribution is only of value if the final peer-rating (expertise) and credibility values output by social-expert are accurate. To measure the level of accuracy we need a "gold standard" representing true expertise and credibility for all nodes. We can then compare these true values to the social-expert calculated values. The

Tests	Initial peer-rating $p(j)_0, j \in N$	Initial credibility $c(j)_0, j \in N$
Test 1	0	0
Test 2	0.1	0.1
Test 3	0.5	0.5
Test 4	0.75	0.75
Test 5	0.9	0.9
Test 6	1	1
Test 7	*random*	*random*
Test 8	*average*	0.5

Table 4.3: Tests with different initial value combinations.

level of success can be determined by measuring how close the social-expert values match the true values in the gold standard.

To formalize the above we denote the "gold standard" as a vector called *standard*. Its ith element , $standard(i)$, is a value pair $p_{TRUE}(i), c_{TRUE}(i)$, where $p_{TRUE}(i)$ and $c_{TRUE}(i)$ represent the true expertise and credibility of node i respectively. We want to compare the output from social-expert to $p_{TRUE}(i)$ and $c_{TRUE}(i)$ (for all nodes $i \in N$). Recall from chapter 3 that the output of social-expert is also a vector (denoted as SR), where the i-th coordinate of the vector is a value pair: $SR(i) = p(i), c(i)$.

To measure the effectiveness of social-expert quantitatively we can calculate the Pearson product moment correlation coefficient (r) between vectors *standard* and SR. We will then end up with two Pearson coefficients, $r_{peer-rating}$ and $r_{credibility}$, representing the correlation between peer-rating and credibility of nodes respectively. In addition to direct comparison of values it may be interesting to produce the ranking or order for the nodes in terms of their relative expertise (peer-rating) and credibility levels. We can then quantitatively compare these orderings in vectors *standard* and SR using the Spearman's rank correlation coefficient (ρ). Since these vectors hold both peer-rating and credibility values for each node, we will end up with two Spearman coefficients, $\rho_{peer-rating}$ and $\rho_{credibility}$.

Having described our approach for measuring output accuracy we can now discuss how to obtain this "gold standard" vector.

4.2.3 Establishing the "gold standard" vector

Ideally we can use human surveyors to produce a standard by interviewing a sample of users from a real online community. For example, Adamic et al. in [25] used interviews to determine expertise levels for a subset of users. Although one of our input datasets, $G_{ADVOGATO}$, represents a real online expertise community, we did not have the resources to interview members. We have considered this as a future improvement for social-expert evaluation. Instead, we use an alternate method as described below.

Recall that for graphs G_{SMALL}, $G_{FACEBOOK}$ and $G_{ADV-SIM}$ we first generated a *true* expertise and credibility value for each node. Then we assigned a weight value (w) to each edge either close to or away from the true expertise of the sink node, based on the true credibility value of the source node. We denoted ratings that were close to the true expertise as *credible* and others as *non–credible*. Therefore, we added expertise ratings to graphs by considering the true simulated peer-rating expertise and credibility values. With this rationale, we also use these simulated true values to populate the *standard* vector.

Note that we are unable to populate the *standard* vector for graph $G_{ADVOGATO}$ using the above method, as we did not simulate its expertise ratings. For this graph, expertise rating information between any two nodes is obtained directly from the Advogato network dataset [29]. Therefore, using the method outlined in this section, we can only perform social-expert output accuracy evaluation for G_{SMALL}, $G_{FACEBOOK}$ and $G_{ADV-SIM}$. However, we still use $G_{ADVOGATO}$ for output precision measurements.

CHAPTER 5
Results

After measuring output precision we were able to confirm that all social-expert calculated scores converged to stable values. Furthermore, these values did not depend on how we chose to initialize graphs: for the same graph, tests with different initial value combinations produced the same final result[1]. In terms of measuring output accuracy we observed that it was quite dependent on the distribution of *credible* and *non–credible* ratings (or edges). As long as about 80% of the edges in the graph were *credible* the algorithm performed well. We also observed that social-expert accuracy was better when executed on graphs that had a higher number of edges per node. We elaborate on these results below.

5.1 Output Precision

To measure output precision we first initialized all input graphs to 0.5. This means each node in the graph was initially assigned 0.5 peer-rating and credibility. We found that after running social-expert, the output peer-rating and credibility score for every node converged as per the convergence criteria described in section 4.2.1.2. As an example, table 5.1 shows the peer-rating values up to 30 iterations for $Node\,0$ in $G_{SMALL\,0.9}$ and the corresponding convergence calculation. The "Difference" column shows that the difference between running average and value at current iteration approached 0 as the number of iterations increased. The same pattern was seen for peer-rating and credibility values after 30 iterations for all graphs and for all nodes in the graphs. This satisfied our convergence criteria:

$$\text{Convergence for } p(j) : lim_{n\to\infty} p(j)_{DIFF,n} = 0.$$
$$\text{Convergence for } c(j) : lim_{n\to\infty} c(j)_{DIFF,n} = 0.$$

Figures 5.1. 5.2 and 5.3 show the maximum difference ($max\ p(i)_{DIFF,n}$ from all $i \in N$) between the running average and value at current iteration. They compare

[1]Only initializing all nodes with 0s did not work as it resulted in a divide by zero error. We elaborate on this in section 5.2.

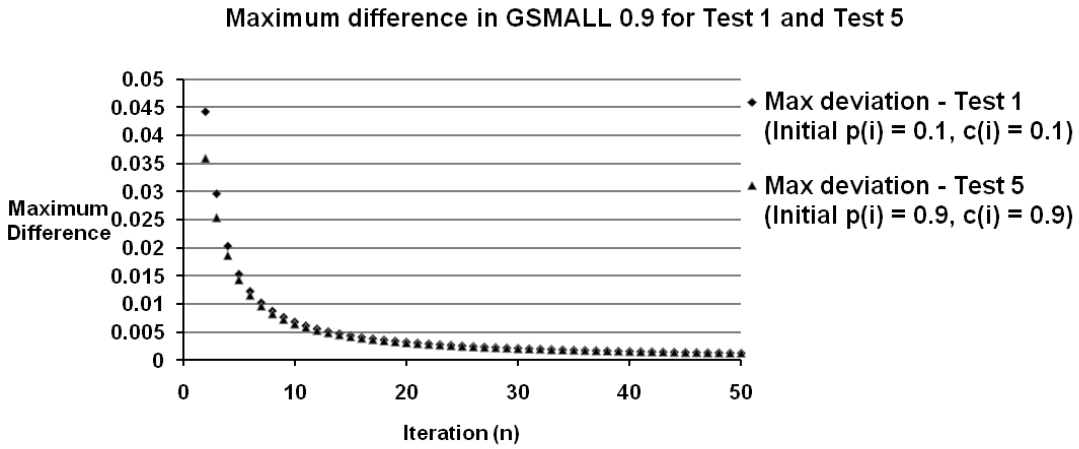

Figure 5.1: *Maximum* difference $(p(i)_{DIFF,n})$ between running average and value at current iteration for $G_{SMALL\,0.9}$ in test 1 $(p(i)_0 = 0.1, c(i)_0 = 0.1, i \in N)$ and test 5 $(p(i)_0 = 0.9, c(i)_0 = 0.9, i \in N)$.

the convergence speed in Test 1 $(p(i)_0 = 0.1, c(i)_0 = 0.1, i \in N)$ and Test 5 $(p(i)_0 = 0.9, c(i)_0 = 0.9, i \in N)$. The fast convergence is clearly seen as the *maximum* difference reaches close to zero very quickly.

Note that although we used $n \to \infty$ in our definition it was actually unnecessary to run a very large number of iterations. We used 17 decimal places in our calculations and found that after $n = 30$ almost all nodes reached stable values (up to 17 decimal places). Once *all* nodes reached stable values, further iterations did not change the output scores for any node. For example, if all nodes reached stable values, say at the 39th iteration $(n = 39)$, then for all iterations $n \geq 39$ the peer-rating and credibility of all nodes stayed constant:

$$\text{For all nodes } j \in N,\ n \geq 39 \to p(j)_n = p(j)_{n+1}$$
$$\text{For all nodes } j \in N,\ n \geq 39 \to c(j)_n = c(j)_{n+1}$$

The example in table 5.1 shows that the initial peer-rating value we assigned to $Node\,0$ (and every other node in the graph) was 0.5. In this example stable values to 17 decimal places were reached for $Node\,0$ by the 20th iteration. Next, we changed this initial value to 0.75 and checked whether after 30 iterations the results are the same as when the graph was initialized to 0.5. Indeed we found this to be the case, thereafter performing 6 more tests with different initial value combinations.

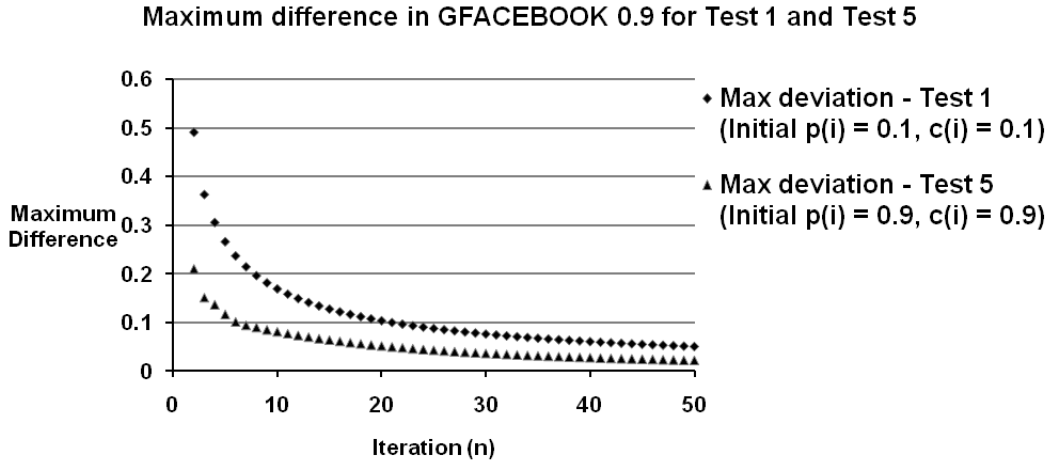

Figure 5.2: ***Maximum*** **difference** $(p(i)_{DIFF,n})$ **between running average and value at current iteration for** $G_{FACEBOOK\,0.9}$ **in test** 1 $(p(i)_0 = 0.1, c(i)_0 = 0.1, i \in N)$ **and test** 5 $(p(i)_0 = 0.9, c(i)_0 = 0.9, i \in N)$**.**

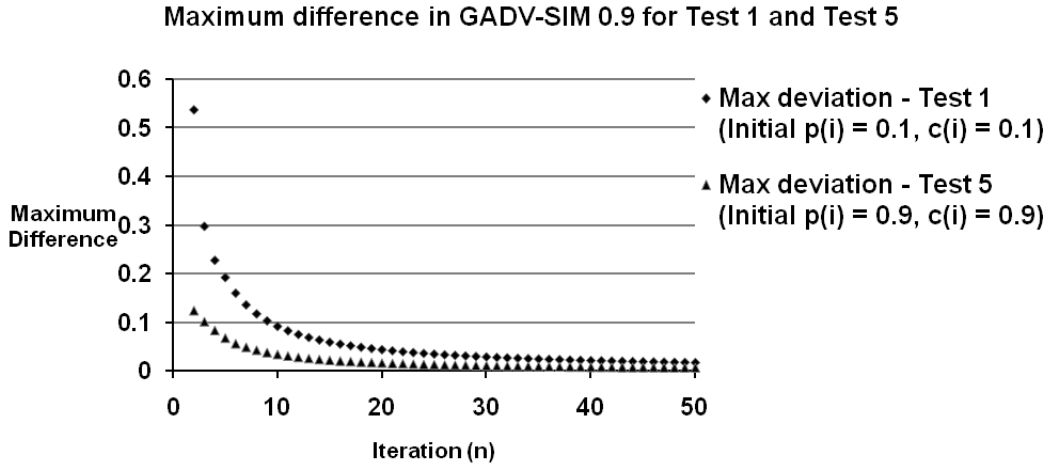

Figure 5.3: ***Maximum*** **difference** $(p(i)_{DIFF,n})$ **between running average and value at current iteration for** $G_{ADV-SIM\,0.9}$ **in test** 1 $(p(i)_0 = 0.1, c(i)_0 = 0.1, i \in N)$ **and test** 5 $(p(i)_0 = 0.9, c(i)_0 = 0.9, i \in N)$**.**

$Node\,0$, $G_{SMALL\,0.9}$			
Iteration (n)	peer-rating ($p(Node\,0)_n$)	Running average ($\overline{p(Node\,0)_{n-1}}$)	Difference
Initial ($n = 0$)	0.5		
1	0.7584688725241916		
2	0.7502101470521465	0.75846887	0.00825872
3	0.7508678685321908	0.75433950	0.00347164
4	0.7510153678117442	0.75318229	0.00216692
5	0.7510390031115917	0.75264056	0.00160156
		...	
18	0.7510433049911849	0.75141883	$3.7552510E-4$
19	0.7510433049911851	0.75139796	$3.5466259E-4$
20	0.7510433049911849	0.75137930	$3.3599614E-4$
21	0.7510433049911849	0.75136250	$3.1919633E-4$
		...	
28	0.7510433049911849	0.75127974	$2.3644173E-4$
29	0.7510433049911849	0.75127130	$2.2799738E-4$
Final ($n = 30$)	0.7510433049911849	0.75126344	$2.2013540E-4$

Table 5.1: Convergence of peer-rating values for $Node\,0$ in $G_{SMALL\,0.9}$.

Table 4.3 lists the initial peer-rating and credibility values corresponding to each test. As an example, tables 5.2 and 5.3 compare the initial and final peer-rating and credibility values across all 8 tests for $Node\,0$ in graph $G_{SMALL\,0.9}$. As shown in these tables for $Node\,0$, we also observed identical final peer-rating and credibility output for all other nodes regardless of how the graph was initialized. Test 1 however was an exception to this; here we initialized each node to have a peer-rating and credibility of 0. As expected this resulted in a divide by zero error when calculating peer-rating as per equation 3.6. Note that this does not mean every edge must have a non-zero rating (in fact our simulations contained many ratings with a 0 weight). It only implies that graphs must be initialized to any value > 0 before social-expert can be executed. As another example, table 5.4 shows a comparison of the final values between test 2 and test 5 for a subset of nodes in $G_{SMALL\,0.9}$.

Lastly, to confirm that the final values for each node were identical across all tests, we ran a series of *t-tests* between the results from each test for all graphs. The p-value received from the t-tests were always 1. This confirmed that there were no

$Node\,0$, $G_{SMALL\,0.9}$ - Peer-rating		
Tests	Initial ($n = 0$) $p(Node\,0)_0$	Final ($n = 30$) $p(Node\,0)_{30}$
Test 1	0	$Divide\,by\,0$
Test 2	0.1	0.7510433049911852
Test 3	0.5	0.7510433049911852
Test 4	0.75	0.7510433049911852
Test 5	0.9	0.7510433049911852
Test 6	1	0.7510433049911852
Test 7 (random)	0.8748605636762333	0.7510433049911852
Test 8 (average)	0.7321477210242879	0.7510433049911852

Table 5.2: Initial and final peer-rating of $Node\,0$ in $G_{SMALL\,0.9}$.

$Node\,0$, $G_{SMALL\,0.9}$ - Credibility		
Tests	Initial ($n = 0$) $c(Node\,0)_0$	Final ($n = 30$) $c(Node\,0)_{30}$
Test 1	0	$Divide\,by\,0$
Test 2	0.1	0.9766226516975955
Test 3	0.5	0.9766226516975955
Test 4	0.75	0.9766226516975955
Test 5	0.9	0.9766226516975955
Test 6	1	0.9766226516975955
Test 7 (random)	0.8748605636762333	0.9766226516975955
Test 8 (average)	0.7321477210242879	0.9766226516975955

Table 5.3: Initial and final credibility of $Node\,0$ in $G_{SMALL\,0.9}$.

statistically significant differences between the final output from these tests. Table 5.5 shows the calculated p-values between each test performed for $G_{ADV-SIM\,0.9}$. We used test 3 as a comparison standard. We did not show comparisons with test 1, since there was no valid output from this test due to the divide by zero error.

Similar to the p-values in table 5.5, we tested and obtained the same results for $G_{SMALL\,0.9}$, $G_{ADVOGATO}$ and $G_{ADV-SIM\,0.9}$. Hence we have proved the null hypothesis: *the final values from social-expert output are not dependent upon the initial values assigned to all nodes in the graph.*

$G_{SMALL\,0.9}$- Peer-rating values		
	Test 2 ($p(j)_0 = 0.1$ and $c(j)_0 = 0.1$) ($j \in N$)	**Test 5** ($p(j)_0 = 0.9$ and $c(j)_0 = 0.9$) ($j \in N$)
Node (j)	Final values ($n = 30$) ($p(j)_{30}$)	Final values ($n = 30$) ($p(j)_{30}$)
Node 0	0.7510433049911849	0.7510433049911849
Node 1	0.2951362342850895	0.2951362342850895
Node 2	0.8754239492770773	0.8754239492770773
Node 3	0.7383863448676096	0.7383863448676096
Node 4	0.2420377876657297	0.2420377876657297
...	...	...
Node 36	0.5075420361757191	0.5075420361757191
Node 37	0.5041359582949425	0.5041359582949425
Node 38	0.5074647500918666	0.5074647500918666
Node 39	0.4674811271707014	0.4674811271707014

Table 5.4: Comparing final peer-rating values in $G_{SMALL\,0.9}$ for test 2 and test 5.

$G_{ADV-SIM\,0.9}$			
Tests compared (for different initial values of $j \in N$)			p-value
Test 2 ($p(j)_0 = 0.1$, $c(j)_0 = 0.1$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1
Test 4 ($p(j)_0 = 0.75$, $c(j)_0 = 0.75$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1
Test 5 ($p(j)_0 = 0.9$, $c(j)_0 = 0.9$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1
Test 6 ($p(j)_0 = 1.0$, $c(j)_0 = 1.0$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1
Test 7 ($p(j)_0 = random$, $c(j)_0 = random$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1
Test 8 ($p(j)_0 = average$, $c(j)_0 = 0.5$)	and	Test 3 ($p(j)_0 = 0.5$, $c(j)_0 = 0.5$)	1

Table 5.5: p-values from t-tests between test outputs for $G_{ADV-SIM\,0.9}$.

5.2 Output Accuracy

Our results show that social-expert is able to determine the true expertise and credibility of nodes reasonably accurately. Furthermore, the level of accuracy is dependent upon the distribution of *credible* and *non–credible* edges in the network.

To quantitatively measure the level of accuracy we calculated the Pearson product moment correlation coefficient ($r_{peer-rating}$) between the simulated true peer-rating expertise level and the social-expert output expertise level. We also calculated this coefficient between the simulated true credibility and social-expert output credibility ($r_{credibility}$). Table 5.6 shows these Pearson coefficients. The correlation coefficient for peer-rating ($r_{peer-rating}$) remained quite high as long as the % of *credible* edges were $\geq$ 80%. Below 80%, correlation began to drop sharply; reducing below 0 as the distribution of *credible* edges approached 50%. Correlation for credibility ($r_{credibility}$) were in general higher than peer-rating. An exception to this was when 100% nodes were *good* and close to 100% edges were *credible* (graphs $G_{SMALL\,1.0}$, $G_{FACEBOOK\,1.0}$ and $G_{ADV-SIM\,1.0}$). We suspect this happened because we used a uniform distribution with a very narrow range for true credibility (section 4.1.4.3 describes this distribution further). Recall that a *good* node always has simulated true credibility between 0.95 and 1.0, while a *bad* node has simulated true credibility between 0 and 0.3. When 100% of the nodes are *good*, the true credibility of *every* node lied in a narrow 0.05 range. We believe this narrow range in turn negatively affected the correlation comparison. As soon as we introduced more *non–credible* edges, the range of true credibility values and $r_{credibility}$ increased. The performance on $G_{ADV-SIM}$ graphs were slightly worse than others. We believe this was due to a higher number of low degree *bad* nodes in the graph; we elaborate on this later in the section.

To test output accuracy further we ordered each node in the network with respect to its true peer-rating and its social-expert calculated peer-rating. We then compared these two node orderings using the Spearman's rank correlation coefficient and obtained $\rho_{peer-rating}$. We made the same comparisons for credibility orderings and obtained $\rho_{credibility}$. Table 5.7 shows these Spearman coefficients. The pattern for Spearman ranking coefficients were similar to Pearson, as long as the % of

Pearson correlation coefficients (r)				
Graph	Percentages (%) *Good* nodes	Percentages (%) *Credible* edges	**Pearson Coefficients** $r_{peer-rating}$	**Pearson Coefficients** $r_{credibility}$
G_{SMALL}:				
$G_{SMALL\,1.0}$	100	98.5	0.9979	0.6981
$G_{SMALL\,0.9}$	85.0	84.5	0.9750	0.9973
$G_{SMALL\,0.8}$	77.5	79.7	0.9612	0.9982
$G_{SMALL\,0.7}$	67.5	72.1	0.5952	0.9968
$G_{SMALL\,0.6}$	60.0	62.4	0.1504	0.9954
$G_{SMALL\,0.5}$	47.5	54.4	−0.3774	0.9690
$G_{FACEBOOK}$:				
$G_{FACEBOOK\,1.0}$	100	99.1	0.9925	0.3330
$G_{FACEBOOK\,0.9}$	90.0	90.0	0.8904	0.9500
$G_{FACEBOOK\,0.8}$	80.0	81.6	0.7200	0.9384
$G_{FACEBOOK\,0.7}$	70.0	72.6	0.4399	0.9082
$G_{FACEBOOK\,0.6}$	60.5	65.3	0.1534	0.8399
$G_{FACEBOOK\,0.5}$	49.3	55.6	−0.1916	0.5136
$G_{ADV-SIM}$:				
$G_{ADV-SIM\,1.0}$	100	99.0	0.98874	0.11013
$G_{ADV-SIM\,0.9}$	90.2	89.8	0.76052	0.75340
$G_{ADV-SIM\,0.8}$	79.4	80.5	0.48311	0.69907
$G_{ADV-SIM\,0.7}$	70.7	71.8	0.27827	0.64230
$G_{ADV-SIM\,0.6}$	60.4	60.8	−0.0454	0.42377
$G_{ADV-SIM\,0.5}$	50.5	55.0	−0.1779	0.22961

Table 5.6: Pearson correlation coefficients between *true* and social-expert *calculated* values.

credible edges remained about 80%, the ranking correlations were strong. However, the Spearman coefficients were in general lower than their Pearson counterparts. Once again the performance on $G_{ADV-SIM}$ graphs were an exception, especially for $\rho_{credibility}$, as no strong correlation were observed in $G_{ADV-SIM}$ graphs for $\rho_{credibility}$.

A trend we noticed for output accuracy across graphs G_{SMALL}, $G_{FACEBOOK}$ and $G_{ADV-SIM}$ is that correlation coefficients were higher for graphs with a higher average degree (more edges per node). Figure 5.4 shows this trend. Note that the x-axis values indicating average degree per node were obtained from table 4.1. This trend is consistent with social-expert design: in its essence, peer-rating and credibility scores are estimates of a node's qualities from the network's perspective.

Spearman's rank correlation coefficients (ρ)				
Graph	Percentages (%) *Good* nodes	Percentages (%) *Credible* edges	**Spearman Coefficients** $\rho_{peer-rating}$	**Spearman Coefficients** $\rho_{credibility}$
G_{SMALL}:				
$G_{SMALL\,1.0}$	100	98.5	0.9975	0.7332
$G_{SMALL\,0.9}$	85.0	84.5	0.9694	0.8735
$G_{SMALL\,0.8}$	77.5	79.7	0.9467	0.7962
$G_{SMALL\,0.7}$	67.5	72.1	0.4597	0.9266
$G_{SMALL\,0.6}$	60.0	62.4	−0.0263	0.8128
$G_{SMALL\,0.5}$	47.5	54.4	−0.4420	0.8495
$G_{FACEBOOK}$:				
$G_{FACEBOOK\,1.0}$	100	99.1	0.9926	0.0897
$G_{FACEBOOK\,0.9}$	90.0	90.0	0.9023	0.3548
$G_{FACEBOOK\,0.8}$	80.0	81.6	0.7466	0.5307
$G_{FACEBOOK\,0.7}$	70.0	72.6	0.4642	0.6331
$G_{FACEBOOK\,0.6}$	60.5	65.3	0.1417	0.6833
$G_{FACEBOOK\,0.5}$	49.3	55.6	−0.2568	0.5261
$G_{ADV-SIM}$:				
$G_{ADV-SIM\,1.0}$	100	99.0	0.98022	−0.12819
$G_{ADV-SIM\,0.9}$	90.2	89.8	0.75631	0.02269
$G_{ADV-SIM\,0.8}$	79.4	80.5	0.45066	0.11819
$G_{ADV-SIM\,0.7}$	70.7	71.8	0.18513	0.18730
$G_{ADV-SIM\,0.6}$	60.4	60.8	−0.2520	0.16007
$G_{ADV-SIM\,0.5}$	50.5	55.0	−0.4330	0.05772

Table 5.7: Spearman's rank correlation coefficients between *true* and social-expert *calculated* values.

Since they are calculated based on peer evaluations, the more evaluations received and made, the more accurate we can expect these estimates to be.

In addition to correlation and ranking coefficients, we were interested in knowing whether credibility percentiles could be used to identify *bad* nodes. This was indeed the case as we noticed that across all graphs *bad* nodes consistently had the lowest credibility percentiles. Table 5.8 shows that $> 98\%$ of *bad* nodes in $G_{FACEBOOK\,0.9}$ had the lowest 10% credibility scores. Interestingly this table also shows 35 *bad* nodes with very high credibility (over 90th percentile). Upon further investigation we found that for these 35 *bad* nodes to receive such high credibility scores, the following two conditions were true. First, they had a very low out-degree;

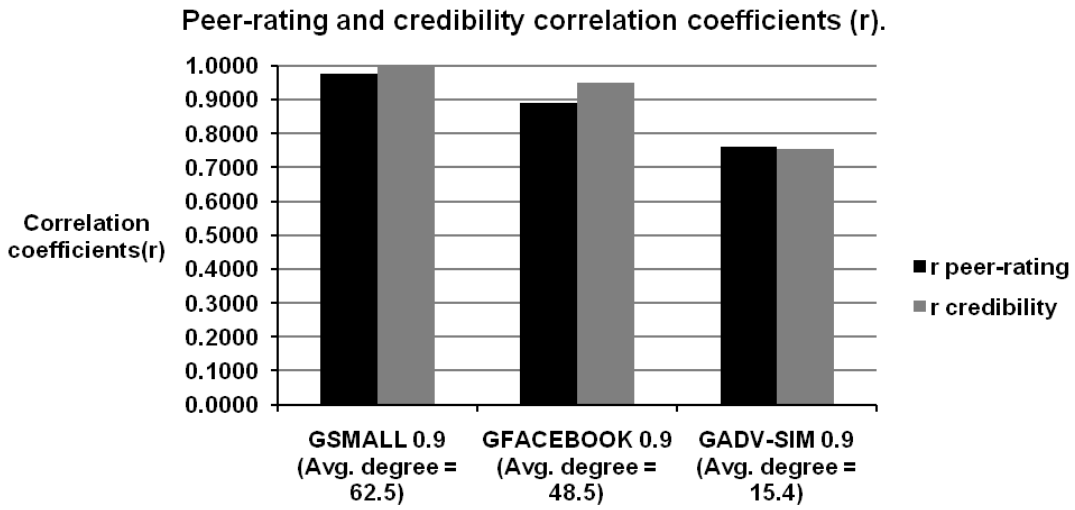

Figure 5.4: Correlation coefficients $r_{peer-rating}$ and $r_{credibility}$ for graphs $G_{SMALL\,0.9}$, $G_{FACEBOOK\,0.9}$ and $G_{ADV-SIM\,0.9}$.

almost all had only one outgoing edge, some had a few more. Second, the nodes they evaluated had very low in-degrees. We refer to such occurrences where both of these conditions were true as: *credibility inflation*. Since when these conditions were true, social-expert incorrectly calculated a very high (and thus, *inflated*) credibility value for a *bad* node. Note that credibility inflation scenarios also resulted in inaccurate peer-rating values for the nodes being evaluated by the *bad* nodes. As an example, let us assume that node i is a *bad* node that only produces one expertise rating, r, and that this rating is made to node j with a *non–credible* edge: $r = (i, j, w)$. Also assume that the only rating node j receives is r. Hence the in-degree of j and out-degree of i are 1: $|OUT(i)| = 1$ and $|IN(j)| = 1$. In such cases social-expert will incorrectly render i as 100% credible. Node i's credibility is calculated based on the difference between node j's peer-rating ($p(j)$) and the weight value (w) of rating r. Since r is the *only* rating that node j receives, $p(j) = w$, resulting in 100% credibility for i. Therefore, this high credibility value can be considered *inflated*, as i being a *bad* node should receive low credibility. Furthermore, the peer-rating of j remains inaccurate since w is from a *non–credible* edge and away from true expertise of j. Figure 5.5 depicts this example, where both $p(j)$ and $c(i)$ are away from their true values due to the one *non–credible* rating with $w = 0.1$. Such in-

$G_{FACEBOOK\,0.9}$ - Credibility percentiles		
Credibility percentile ranges $c(j), j \in N$	Number of nodes *good*	*bad*
$0\% \leq c(j) < 10\%$	51	5960
$10\% \leq c(j) < 20\%$	5973	37
$20\% \leq c(j) < 30\%$	6010	0
$30\% \leq c(j) < 40\%$	6010	0
$40\% \leq c(j) < 50\%$	6010	0
$50\% \leq c(j) < 60\%$	6011	0
$60\% \leq c(j) < 70\%$	6010	0
$70\% \leq c(j) < 80\%$	6010	0
$80\% \leq c(j) < 90\%$	6010	0
$90\% \leq c(j) < 100\%$	5975	35

Table 5.8: Number of *good* and *bad* nodes in $G_{FACEBOOK\,0.9}$ credibility percentile ranges.

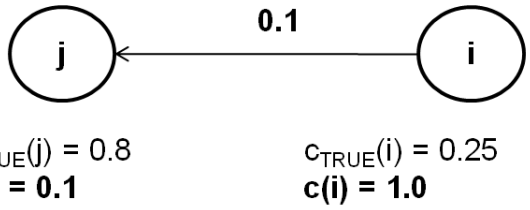

Figure 5.5: Credibility inflation scenario example with 2 nodes.

correct values can be prevented if j receives some *credible* ratings from other nodes or if i performs additional peer ratings. In summary, social-expert relies on peer evaluations to estimate a node's true expertise. Hence, the more *credible* ratings a node receives the better the estimate. An inaccurate peer-rating results when a node receives only *non–credible* ratings (or a very high number of *non–credible* ratings). In addition, if the node receives its *non–credible* rating from a peer node with very low out-degree, a credibility inflation scenario occurs where social-expert incorrectly estimates very high credibility for that peer. If the peer performs more ratings, social-expert can produce a better estimate of its true credibility.

Inflated credibility of *bad* nodes with low out-degrees did not occur in G_{SMALL} as it was a very well connected graph with *every* node having many incoming and outgoing edges. Hence the Pearson and Spearman coefficients are also higher in the G_{SMALL} graph versions. Of course this type of topology where all nodes are

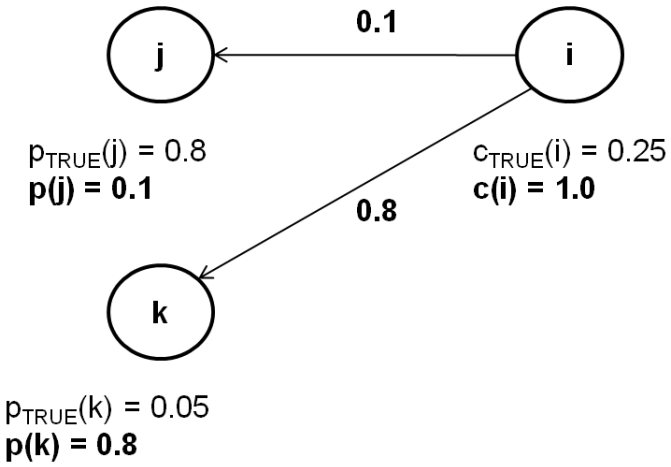

Figure 5.6: Credibility inflation scenario example with 3 nodes.

uniformly connected to many other nodes does not reflect naturally occurring networks (please see section 4.1.3 for more details). Graphs from real online networks, $G_{FACEBOOK}$ and $G_{ADV-SIM}$, contained many cases of small nodes groups with ≤ 5 nodes where in general each node had low in-degree and out-degree. We observed inflated credibility values in *bad* nodes when they were present in such groups. This happened much more in the $G_{ADV-SIM}$ graphs; which we suspect was the reason why $G_{ADV-SIM}$ correlation and ranking coefficients were lower than $G_{FACEBOOK}$ graphs. To confirm this suspicion we created additional graphs where we detected and removed instances of groups with two or three nodes having at least one *bad* node with low out-degree. More specifically, we detected the condition where a node i with an out-degree of 1 or 2 ($|OUT(i)| \leq 2$) was placing *non–credible* ratings to one or two other nodes, j and k. Also, the only ratings nodes j and k received were from node i: $|IN(j)| = 1$ and $|IN(k)| = 1$. An example of this scenario with three nodes is shown in figure 5.6. We then eliminated such inflated credibility scenarios by rendering node i as a *good* node. To compensate for *bad* nodes being converted into *good* nodes, we also converted a set of randomly selected *good* nodes (with > 2 ratings) to *bad* nodes. However, we ensured that the distribution of *credible* and *non–credible* ratings in the modified graph was approximately the same as the original graph.

Tables 5.9 and 5.10 show Pearson and Spearman coefficients for graphs with

Pearson correlation coefficients (r)			
Graphs	Inflation instances removed	$r_{peer-rating}$	$r_{credibility}$
$G_{FACEBOOK}$ and $G^*_{FACEBOOK}$			
$G_{FACEBOOK\,0.9}$	138	0.8904	0.9500
$G^*_{FACEBOOK\,0.9}$		0.9030	0.9564
$G_{FACEBOOK\,0.8}$	263	0.7200	0.9384
$G^*_{FACEBOOK\,0.8}$		0.7277	0.9445
$G_{FACEBOOK\,0.7}$	387	0.4399	0.9082
$G^*_{FACEBOOK\,0.7}$		0.4800	0.9154
$G_{ADV-SIM}$ and $G^*_{ADV-SIM}$			
$G_{ADV-SIM\,0.9}$	173	0.7605	0.7534
$G^*_{ADV-SIM\,0.9}$		0.9050	0.9291
$G_{ADV-SIM\,0.8}$	332	0.4831	0.6991
$G^*_{ADV-SIM\,0.8}$		0.7438	0.9012
$G_{ADV-SIM\,0.7}$	537	0.2783	0.6423
$G^*_{ADV-SIM\,0.7}$		0.4886	0.8472

Table 5.9: Comparison of Pearson correlation coefficients between original graphs ($G_{FACEBOOK}$ and $G_{ADV-SIM}$) and graphs with some credibility inflation scenarios removed ($G^*_{FACEBOOK}$ and $G^*_{ADV-SIM}$).

approximately the same number of *credible* edges that underwent the above procedure. We denote these modified graphs as: $G^*_{FACEBOOK}$ and $G^*_{ADV-SIM}$. We see that these graphs have higher coefficients than their original counterparts. We also notice that the "inflation instances removed" column has higher numbers for $G^*_{ADV-SIM}$ than $G^*_{FACEBOOK}$. These observations confirm 2 suspicions: credibility inflation indeed affected social-expert performance and $G_{ADV-SIM}$ graphs contained higher levels of credibility inflation scenarios, resulting in lower Pearson and Spearman coefficients. We believe that the higher number of inflation scenarios in $G_{ADV-SIM}$ graphs were due to the inherent topology of the Advogato dataset. Table 4.1 shows that $G_{ADV-SIM}$ has a reasonably lower average degree per node than $G_{FACEBOOK}$. This implies that it is not as well connected and hence is likely to have more small node groups with very low degrees where credibility inflation can be simulated as in the example scenarios of figures 5.5 and 5.6.

Table 5.11 shows the credibility percentiles for graph $G^*_{FACEBOOK\,0.9}$. We see

Spearman ranking correlation coefficients (ρ)			
Graphs	Inflation instances removed	$\rho_{peer-rating}$	$\rho_{credibility}$
$G_{FACEBOOK}$ and $G^*_{FACEBOOK}$			
$G_{FACEBOOK\,0.9}$		0.9023	0.3548
$G^*_{FACEBOOK\,0.9}$	138	0.9133	0.3551
$G_{FACEBOOK\,0.8}$		0.7466	0.5307
$G^*_{FACEBOOK\,0.8}$	263	0.7493	0.5355
$G_{FACEBOOK\,0.7}$		0.4642	0.6331
$G^*_{FACEBOOK\,0.7}$	387	0.4942	0.6395
$G_{ADV-SIM}$ and $G^*_{ADV-SIM}$			
$G_{ADV-SIM\,0.9}$		0.7563	0.0227
$G^*_{ADV-SIM\,0.9}$	173	0.8986	0.1110
$G_{ADV-SIM\,0.8}$		0.4507	0.1182
$G^*_{ADV-SIM\,0.8}$	332	0.7364	0.2642
$G_{ADV-SIM\,0.7}$		0.1851	0.1873
$G^*_{ADV-SIM\,0.7}$	537	0.4555	0.4005

Table 5.10: Comparison of Spearman ranking coefficients between original graphs ($G_{FACEBOOK}$ and $G_{ADV-SIM}$) and graphs with some credibility inflation scenarios removed ($G^*_{FACEBOOK}$ and $G^*_{ADV-SIM}$).

that 5 *bad* nodes still incorrectly appear with very high credibility. However, this amount is greatly reduced compared to the 35 *bad* nodes in the original graph where credibility inflation scenarios not removed. We found that these remaining 5 *bad* nodes with high credibility were instances of credibility inflation with groups larger than just 2 or 3 nodes. We only removed inflation instances with groups containing ≤ 3 nodes.

$G^*_{FACEBOOK\,0.9}$ - Credibility percentiles		
Credibility percentile ranges $c(j), j \in N$	Number of nodes *good*	*bad*
$0\% \leq c(j) < 10\%$	59	5952
$10\% \leq c(j) < 20\%$	5998	12
$20\% \leq c(j) < 30\%$	6010	0
$30\% \leq c(j) < 40\%$	6010	0
$40\% \leq c(j) < 50\%$	6010	0
$50\% \leq c(j) < 60\%$	6011	0
$60\% \leq c(j) < 70\%$	6010	0
$70\% \leq c(j) < 80\%$	6010	0
$80\% \leq c(j) < 90\%$	6010	0
$90\% \leq c(j) < 10\%$	6005	5

Table 5.11: Number of *good* and *bad* nodes in $G^*_{FACEBOOK\,0.9}$ credibility percentile ranges (with some credibility inflation instances removed).

CHAPTER 6
Discussion and Future Work

In the previous chapter we described our results from testing social-expert in terms of convergence to stable values and ability to determine nodes' expertise effectively. From our test results we found that social-expert outputs converged for 100% of the tests we ran. We observed that social-expert effectiveness varied based on the % of *credible* edges and on the average number of peer evaluations made and received per node. We also discovered that it was easy to manipulate one's credibility by having a low out-degree, which we denoted as credibility inflation. In this chapter we discuss some implications of these results.

We observed that Pearson and Spearman correlation coefficients for social-expert calculated scores decreased as the % of *non–credible* edges increased. The distribution of *credible* to *non–credible* edges were in turn based on the % of *good* and *bad* nodes (i.e. more good nodes resulted more *credible* edges). When interpreting this finding it is important to note that a node being *good* or *bad* is not dependent upon its topic expertise. Recall that social-expert considers each node's peer-rating and credibility qualities independent of each other. The peer-rating score represents a node's "topic expertise". The credibility score represents a node's "social expertise", in that, it reflects his or her expertise or knowledge of the society. For example, a node with high expertise who is however a poor judge of others' expertise should have a high peer-rating and a low credibility score. In our tests we simulated topic expertise with a Gaussian distribution, while we varied credibility with uniform distributions of *good* and *bad* nodes.

We found that graphs having at least 80% *good* nodes resulted strong correlation. What does this finding imply for organizations that could utilize social-expert? One implication is that in general social-expert should be run on networks where we can expect about 4 out 5 ratings to be credible. Whether this expectation is reasonable depends on how experienced the nodes are with the network. For example, Ackerman and McDonald mention that individuals with the "*expertise concierge*"

role were "*technically sophisticated, had relatively long tenure with the organization and had high-status positions*" [1]. These are qualities that one typically develops with experience. Such nodes should have the highest credibility scores. Conversely, the newer inexperienced nodes are likely to have low credibility, as their social maps within the organization are still developing. We can further qualify the statement that 4 out 5 ratings should be credible. A credible rating means that the weight value of the edge is *somewhat* close to the true expertise of the node being rated. The credibility calculation of social-expert inherently allows for bias or variation of opinions on a node's true expertise. The credibility of the rater is only significantly (negatively) affected if the weight value is *quite* far from the true value. Thus, in general ratings do not need to be spot on, or even within a very narrow range of true expertise. Please refer to figure 3.1 which shows how social-expert calculates credibility by allowing variation.

We think that the peer-rating values themselves should be interpreted carefully for nodes with low degrees. As part of measuring output precision we confirmed that how we choose to initialize a graph does not affect the final output of social-expert. Hence we need not worry about producing a reasonable initial estimate for nodes. However, the cold start problem still exists for new nodes that join the network. For social-expert to calculate a node's peer-rating and credibility it needs to have at least one incoming rating ($|IN| > 0$) and at least one outgoing rating ($|OUT| > 0$) respectively. A new node may be missing one of these conditions, implying that we cannot use social-expert to determine either its peer-rating or credibility. Furthermore, because social-expert relies on peer-evaluations to determine the expertise level, nodes that receive more evaluations are likely to have a more accurate peer-rating score. Similarly, the more peer-evaluations a node has performed, the more accurate its credibility score will be. This implies that the accuracy of social-expert for nodes with low in or out degrees (such as newly joined nodes) will be lower than the more well connected ones.

Based on the above observation we may be able to improve social-expert by introducing a third component that we will call *confidence*. A node's peer-rating and credibility confidence would be based on its in-degree ($|IN|$) and out-degree ($|OUT|$)

respectively. A low confidence peer-rating would indicate that we did not receive many peer evaluations and hence the social-expert calculated peer-rating may not be accurate. An important implication would be that peer-rating and credibility scores calculated for nodes involved in credibility inflation scenarios would have low confidence. Such information would be useful for the person performing expertise selection[1]. Confidence upon a node's credibility is more interesting and can be used to potentially improve social-expert's effectiveness. As an example, let us consider two nodes, i and j, who have equally high credibility values: $c(i) = 0.90$, $c(j) = 0.90$. Also let us assume node i has made 40 evaluations, $|OUT(i)| = 40$, while node j has only 2 evaluations $|OUT(j)| = 2$. In such cases the credibility of node i should be more accurate than j since $|OUT(i)| > |OUT(j)|$. In the proposed improvement we would attach a higher confidence weight to $c(i)$. Consequently, when calculating a node's peer-rating based on node i and node j's evaluations, node i's evaluations would be weighted more. In the current design social-expert would place equal weights on evaluations made by i and j.

We observed that it was reasonably easy for nodes to manipulate their credibility and peer-rating scores. For example, nodes within a clique can rate each other highly. As long as the clique is isolated or connected to the remaining graph via a very small number of edges, the artificial values will remain uncorrected. The term "collusion" can be used to describe such clique activity of artificial manipulation or enhancement of scores. The inflation example scenarios shown in figures 5.5 and 5.6 can be considered to be a form of micro collusion (since it involves enhancing scores with only 1 or 2 nodes). We already discussed an improvement to identify potential inflation cases via confidence. One can also argue that because the end goal of social-expert is to produce expertise ratings, it may be unnecessary to introduce much collusion prevention measures. The rationale is that a rich thread of research already exists in the area of collusion prevention through reputation and trust metrics; as discussed below.

Social-expert can be used on a network as a complement to an existing security metric which has already filtered out untrustworthy nodes. For example, the

[1]Expertise selection: given a group of people within the desired expertise range, choosing which individual is right for a task. Please see chapter 1 for more details.

Advogato community calculates a global trust metric for each node as proposed by Levien in his original (unfinished) PhD Thesis [13]. Levien summarizes the application of the Advogato trust metric in online networks as a means for excluding "*trolls, spam, and other forms of abuse common to bulletin board type systems*" [13]. As discussed earlier in section 4.1.4.5, Advogato certifications are done at levels observer, journeyer, apprentice and master. The idea of social-expert being complementary means it could be utilized on the Advogato community as a way to measure expertise only on nodes which are certified as "journeyer" or above. We suggest this as a future work in the development of social-expert. Another important work in this area is the trust inference model [9] of Golbeck and Hendler. This model was implemented in TrustMail, an e-mail system which filters out junk e-mail based on trustworthiness of the sender. Caverlee et al. proposed the Socialtrust algorithm [12] with the purpose of detecting social spam (targeted malware, corrupt user-generated tags) and deception (impersonated digital identities, social network enhanced phishing). Trust metrics have also been proposed as a method to filter malicious nodes in mobile device networks [8], P2P networks [7, 6], and the Web [42]. We refer the interested reader to a widely cited comprehensive survey [10] by Sabater and Sierra on trust and reputation models. In this survey the authors define three categories to describe existing trust and reputation models on node behavior assumptions:

> "*Level 0. Cheating behavior is not considered. The model relies on a large number of agents [nodes] who offer honest ratings to counteract the potential effect of the ratings provided by malicious agents. Level 1. The model assumes that agents can hide or bias the information but they never lie. Level 2. The model has specific mechanisms to deal with liars*" [10].

Although social-expert is not a trust model, the same concept of "lying" about others' expertise levels applies. On the above scale social-expert's assumptions align with levels 0 and 1. We expect that in general people in organizations or communities will not purposefully lie about others' expertise since there is no reward in doing so.

In fact, for producing more accurate ratings, social-expert will inherently reward an individual with a higher credibility score. Based on this, one can argue that it is reasonable to expect of about 80% evaluations in a network to be somewhat accurate.

Another important finding from our study is that social-expert is more effective in graphs where average connectivity per node is higher. This is consistent with social-expert design: the more evaluations a node receives or makes, the more accurate we expect his or her peer-rating and credibility scores to be. For organizations or communities that can utilize social-expert, this implies that members should be encouraged to be more involved with the network (by making or receiving more peer evaluations).

We end our discussion by suggesting a future work of developing a web interface which supports explicit expertise ratings. An important factor to consider with such an interface is that exact peer-rating values should not be publicly displayed. By knowing someone's exact peer-rating a node can artificially enhance its own credibility. The authors of [20] point out another important observation that must be considered:

> *"Explicit rating schemes are unlikely to work for expertise recommendation, because people are reticent to explicitly state opinions of coworkers without an appropriate context. Ratings may not work but it might be possible to collect feedback"* [20].

To accommodate the above insight, peer-evaluation scores must be kept private as well. It would be time consuming to have every user evaluate other users on all possible topics, as there may be many topics of interest in an organization. Hence peer-evaluations should occur through a feedback mechanism. For example, when two individuals have worked with each other on a certain topic, they should be encouraged to enter (private) evaluations for one another. A similar feedback mechanism is proposed in [6] for P2P nodes.

CHAPTER 7
Conclusions

We introduced a ranking algorithm called social-expert that is designed for expertise location in organizations and online communities. Social-expert can be run on network graphs representing a special type of social network that we denote as an "expertise network". Social-expert views each person as having two qualities: peer-rating and credibility. Peer-rating reflects one's expertise on a certain topic and credibility represents his or her aptitude in evaluating others on that topic. Peer-rating is calculated from peer evaluations that a person has received while credibility is calculated from peer evaluations that he or she has made. We tested social-expert on a set of expertise network graphs constructed from a mixture of simulated and real world data. We found that social-expert outputs converged relatively quickly with most nodes' scores reaching stable values up to 17 decimal places within 30 iterations. Our results showed that the effectiveness of social-expert varied according to the general peer-evaluation accuracy. As long as about 80% of peer evaluations were somewhat correct, social-expert was effective in accurately calculating peer-rating and credibility scores. We also found that social-expert was more effective for nodes that made or received more peer ratings. Similarly, peer-rating and credibility scores were not as accurate when a node made or received only one or two ratings and each rating was inaccurate. Thus, by maintaining a low number of total peer evaluations, a node may attempt to manipulate its scores. To ameliorate this we proposed an improvement of adding a confidence value for the scores calculated by social-expert. Individuals or small cliques trying to perform collusion will typically have smaller number of peer evaluations and thus lower confidence scores. People with higher numbers of peer-evaluations will have higher confidence scores. In conclusion, we have validated social-expert as an effective method of finding expertise. Future work remains to be done to test the proposed improvement, compare social-expert performance against other graph ranking algorithms and research how organizations can effectively collect peer-evaluation data.

LITERATURE CITED

[1] David W McDonald and Mark S Ackerman, "Just talk to me: a field study of expertise location," in *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, 1998, pp. 315-324.

[2] Stanley Wasserman and Katherine Faust, *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*, Cambridge: Cambridge University Press, 1994.

[3] Raph Levien. "Advogato," *Advogato*, 1999. [Online]. Available: http://advogato.org/. [Date Last Accessed, 04/09/2010].

[4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, Tech. Rep., 1999.

[5] Jon M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604-632, September 1999.

[6] Li Xiong and Ling Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843-857, July 2004.

[7] Mudhakar Srivatsa, Li Xiong, and Ling Liu, "TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 422-431.

[8] Daniele Quercia, Stephen Hailes, and Licia Capra, "Lightweight Distributed Trust Propagation," in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, 2007, pp. 282-291.

[9] Jennifer Golbeck and James Hendler, "Inferring binary trust relationships in Web-based social networks," *ACM Transactions on Internet Technology (TOIT)*, vol. 6, no. 4, pp. 497-529, November 2006.

[10] Jordi Sabater and Carles Sierra, "Review on Computational Trust and Reputation Models," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33-60, September 2005.

[11] Jennifer A Golbeck, "Computing and Applying trust in Web-Based Social Networks," PhD. thesis, University of Maryland, 2005.

[12] James Caverlee, Ling Liu, and Steve Webb, "Socialtrust: tamper-resilient trust establishment in online communities," in *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, 2008, pp. 104-114.

[13] Levien Raphael, "Attack Resistant Trust Metrics - Ongoing PhD thesis," July 2004. [Online]. Available: http://www.levien.com/thesis/compact.pdf. [Date Last Accessed, 04/09/2010].

[14] L. Streeter and K. Lochbaum, "Who Knows: A System Based on Automatic Representation of Semantic Structure," in *Proceedings of RIAO*, 1988, pp. 380-388.

[15] David W. McDonald, "Evaluating expertise recommendations," in *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, 2001, pp. 214-223.

[16] Adriana Vivacqua and Henry Lieberman, "Agents to assist in finding help," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2000, pp. 65-72.

[17] David Mattox, Mark T Maybury, and Daryl Morey, "Enterprise expert and knowledge discovery," in *Proceedings of the 8th International Conference on Human-Computer Interaction (HCI International '99)*, 1999, pp. 303-307.

[18] Bruce Krulwich and Chad Burkey, "The ContactFinder agent: Answering bulletin board questions with referrals," in *Proceedings of the National Conference on Artificial Intelligence*, 1996, pp. 10-15.

[19] Mark S Ackerman and David W McDonald, "Answer Garden 2: merging organizational memory with collaborative help," in *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, 1996, pp. 97-105.

[20] David W McDonald and Mark S Ackerman, "Expertise recommender: a flexible recommendation system and architecture," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000, pp. 231-240.

[21] Christopher S Campbell, Paul P Maglio, Alex Cozzi, and Byron Dom, "Expertise Identification using Email Communications," *in Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 528-531.

[22] Leonard N Foner, "Yenta: a multi-agent, referral-based matchmaking system," in *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 301-307.

[23] Henry Kautz, Bart Selman, and Mehul Shah, "Referral Web: combining social networks and collaborative filtering," *Communications of the ACM*, vol. 40, no. 3, pp. 63-65, March 1997.

[24] Byron Dom, Iris Eiron, Alex Cozzi, and Yi Zhang, "Graph-based ranking algorithms for e-mail expertise analysis," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 42-48.

[25] Lada Adamic, Mark S Ackerman, and Jun Zhang, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 221-230.

[26] Lada A Adamic, Jun Zhang, Eytan Bakshy, and Mark S Ackerman, "Knowledge Sharing and Yahoo Answers: Everyone Knows Something," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 665-674.

[27] Andreas Paepcke, "Information needs in technical work settings and their implications for the design of computer tools," *Computer Supported Cooperative Work (CSCW)*, vol. 5, no. 1, pp. 63-92, March 1996.

[28] Alan Mislove, Bimal Viswanath, Meeyoung Cha, and Krishna P. Gummadi, "On the evolution of user interaction in Facebook," in *Proceedings of the 2nd ACM workshop on Online social networks*, 2009, pp. 37-42.

[29] Raph Levien, "Advogato - Graph," *Advogato*, 1999. [Online]. Available: http://www.advogato.org/person/graph.dot. [Date Last Accessed, 04/09/2010].

[30] "Welcome to Facebook," *Facebook*, 2005 [Online]. Available: http://www.facebook.com. [Date Last Accessed, 04/09/2010].

[31] Raph Levien and Alexander Aiken, "Attack-resistant trust metrics for public key certification," in *Proceedings of the 7th USENIX Security Symposium*, 1998, pp. 229-242.

[32] Cai-Nicolas Ziegler and Georg Lausen, "Spreading Activation Models for Trust Propagation," in *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, 2004, pp. 83-97.

[33] Stanley Milgram, "The small world problem," *Psychology Today*, vol. 2, no. 60, pp. 60-67, May 1967.

[34] Duncan J. Watts and Steven H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440-442, June 1998.

[35] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 29-42.

[36] Lada Adamic, Orkut Buyukkokten, and Eytan Adar, "A social network caught in the Web," *First Monday*, vol. 8, no. 6, June 2003.

[37] Lun Li, David Alderson, John C. Doyle, and Walter Willinger, "Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications," *Internet Mathematics*, vol. 2, no. 4, pp. 431-523, 2006.

[38] Jon Kleinberg, "The small-world phenomenon: an algorithm perspective," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 2000, pp. 163-170.

[39] Raph Levien, "Advogato - Trust Certifications," *Advogato*, 1999. [Online]. Available: http://advogato.org/certs.html. [Date Last Accessed, 04/09/2010].

[40] Raph Levien, "Advogato - Trust Metric," *Advogato*, 1999. [Online]. Available: http://advogato.org/trust-metric.html. [Date Last Accessed, 04/09/2010].

[41] Paolo Massa and Kasper Souren, "Trustlet, Open Research on Trust Metrics," in *Proceedings of the 2nd Workshop on Social Aspects of the Web (SAW 2008)*, 2008, pp. 31-43.

[42] Zoltn Gyngyi, Hector Garcia-Molina, and Jan Pedersen, "Combating web spam with trustrank," in *Proceedings of the Thirtieth international conference on Very large data bases*, 2004, pp. 576-587.

APPENDIX A
Advogato certification definitions

The following is an excerpt from [39] showing the definition of various certification levels in Advogato.

"***Master***

A Master is the principal author or hard-working co-author of an 'important' free software project, i.e. one that many people depend on, or one that stands out in quality. A Master has command of the tools and is an excellent programmer. Generally, a Master works equivalent to full time (or more) on free software. Ideally, a Master writes clearly about the work and its broader context, and serves as a mentor to others in the free software community.

Journeyer

Journeyers are the people who make free software happen. A journeyer contributes significantly to an important free software project, or is the author of a useful or technically innovative project. A Journeyer is generally a competent programmer, but significant contributions of documentation, artwork, or other non-code goodies counts too. Ideally, a Journeyer works with others in the free software community to polish and refine the library of free software. While not necessarily the equivalent of full time, a Journeyer spends a significant amount of time on free software.

Apprentice

An apprentice is someone who has contributed in some way to a free software project, but is still striving to acquire the skills and standing in the community to make more significant contributions. Ideally, the

Apprentice is in touch with either an individual mentor or a community that helps to gain these skills. An Apprentice spends a significant amount of time learning the craft of software development, whether by hands-on practice, academic study, or careful observation.

Observer

A user with no trust certification is referred to as an observer. This is the default trust level of a new user and the level to which you would certify someone to remove an existing trust certification. An observer does not have posting priviledges and other powers associated with trust users. They have read-only access to Advogato, with the exceptions of editing their own user profile and posting blog entries" [39].