QLectives – Socially Intelligent Systems for Quality
Project no. 231200

Instrument: Large-scale integrating project (IP)
Programme: FP7-ICT

Deliverable D.1.3.1

Simulations of quality emergence

Submission date: 2010-03-01

Start date of project: 2009-03-01 Duration: 48 months

Organisation name of lead contractor for this deliverable:
University of Fribourg

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidental, only for members of the consortium (including the Commission Services)	

Document information

1.1 Author(s)

Author	Organisation	E-mail
Matúš Medo	University of Fribourg	matus.medo@unifr.ch

1.2 Other contributors

Name	Organisation	E-mail
Giulio Cimini	Univestity of Fribourg	giulio.cimini@unifr.ch
Nigel Gilbert	University of Surrey	n.gilbert@surrey.ac.uk
Stanislao Gualdi	University of Fribourg	stanislao.gualdi@unifr.ch

1.3 Document history

Version#	Date	Change
V0.1	15 January, 2010	Starting version, template
V0.2	19 January, 2010	Partial draft available
V0.3	26 January, 2010	First draft completed
V0.4	18 February, 2010	Stylistic revisions
V1.0	01 March, 2010	Approved version to be submitted to EU
V1.1	01 April, 2010	Fixed Bibliography

1.4 Document data

Keywords	quality, reputation, ranking systems, agent-based modeling
Editor address data	matus.medo@unifr.ch
Delivery date	01-03-2010

1.5 Distribution list

Date	Issue	E-mail
	Consortium members	QLECTIVES@list.surrey.ac.uk
	Project officer	Jose.FERNANDEZ-VILLACANAS@ec.europa.eu
	EC archive	INFSO-ICT-231200@ec.europa.eu

QLectives Consortium

This document is part of a research project funded by the ICT Programme of the Commission of the European Communities as grant number ICT-2009-231200.

University of Surrey (Coordinator)
Department of Sociology/Centre
for Research in Social Simulation
Guildford GU2 7XH
Surrey
United Kingdom
Contact person: Prof. Nigel Gilbert
E-mail: n.gilbert@surrey.ac.uk

University of Fribourg
Department of Physics
Fribourg 1700
Switzerland
Contact person: Prof. Yi-Cheng Zhang
E-mail: yi-cheng.zhang@unifr.ch

Technical University of Delft
Department of Software Technology
Delft, 2628 CN
Netherlands
Contact Person: Dr Johan Pouwelse
E-mail: j.a.pouwelse@tudelft.nl

University of Warsaw
Faculty of Psychology
Warsaw 00927
Poland
Contact Person: Prof. Andrzej Nowak
E-mail: nowak@fau.edu

ETH Zurich
Chair of Sociology, in particular
Modelling and Simulation
Zurich, CH-8092
Switzerland
Contact person: Prof. Dirk Helbing
E-mail: dhelbing@ethz.ch

Centre National de la Recherche Scientifique, CNRS
Paris 75006,
France
Contact person: Dr. Camille ROTH
E-mail: camille.roth@polytechnique.edu

University of Szeged
MTA-SZTE Research Group on
Artificial Intelligence
Szeged 6720, Hungary
Contact person: Dr Mark Jelasity
E-mail: jelasity@inf.u-szeged.hu

Institut für Rundfunktechnik GmbH
Munich 80939
Germany
Contact person: Dr. Christoph Dosch
E-mail: dosch@irt.de

QLectives introduction

QLectives is a project bringing together top social modelers, peer-to-peer engineers and physicists to design and deploy next generation self-organising socially intelligent information systems. The project aims to combine three recent trends within information systems:

- **Social networks** - in which people link to others over the Internet to gain value and facilitate collaboration
- **Peer production** - in which people collectively produce informational products and experiences without traditional hierarchies or market incentives
- **Peer-to-Peer systems** - in which software clients running on user machines distribute media and other information without a central server or administrative control

QLectives aims to bring these together to form Quality Collectives, i.e. functional decentralised communities that self-organise and self-maintain for the benefit of the people who comprise them. We aim to generate theory at the social level, design algorithms and deploy prototypes targeted towards two application domains:

- **QMedia** - an interactive peer-to-peer media distribution system (including live streaming), providing fully distributed social filtering and recommendation for quality
- **QScience** - a distributed platform for scientists allowing them to locate or form new communities and quality reviewing mechanisms, which are transparent and promote quality

The approach of the QLectives project is unique in that it brings together a highly inter-disciplinary team applied to specific real world problems. The project applies a scientific approach to research by formulating theories, applying them to real systems and then performing detailed measurements of system and user behaviour to validate or modify our theories if necessary. The two applications will be based on two existing user communities comprising several thousand people - so-called ”Living labs”, media sharing community tribler.org; and the scientific collaboration forum EconoPhysics.

Executive summary

Quality is the cornerstone of the QLectives project which aims to form Quality Collectives: self-organised decentralised communities. However, besides the quality of communities' members, it is also the quality of the created or shared content what is of great importance for the project. In the context of the project, quality is not a given and extrinsic attribute but rather a consensual outcome of evaluations or opinions given by the community's members. The task of information aggregation (or, in general, the management of information overload) is the focus of information filtering (Belkin, 2000; Hanani et al., 2001).

In this deliverable we focus on how to discover and promote high quality content in the ever-changing world where new potentially relevant objects constantly appear, lose their relevance (quality), and disappear. Breaking news of today become ordinary news tomorrow and unimportant news in one week. It is hence important to study algorithms that are not only able to find relevant objects but are able to do it soon, sooner than objects' relevance becomes obvious or even foregone.

Despite the extensive recent development of reputation and recommender systems (Resnick et al., 2000; Herlocker et al., 2004; Witten and Frank, 2005; Hastie et al., 2009), the role of time has been mostly neglected until now. Data mining algorithms usually focus on "who rated what and how" and ignore the time when the rating was given. However, considering time can bring new aspects which would not be possible otherwise: for example, it can help to distinguish an early "discoverer" of quality content from "followers" who merely join already existing trends. It was recently

shown that time-including algorithms may be more resistant to spammers than standard ones (Noll et al., 2009). As pointed out in (Crane and Sornette, 2008), analysis of time evolution of object's popularity itself can also help to distinguish quality content from rubbish.

We study a case where users differ in their ability of recognizing high-quality objects and hence opinion of some of them is inherently more valuable than opinion of others. This is a standard situation and there are some standard algorithms are already known to fit it well (Kleinberg, 1999). Yet, we show that the inclusion of time may considerably improve the obtained results. Then we make the problem more complex by assuming that relevance of all objects decays with time and new objects appear constantly. In result we obtain a highly dynamic system where objects appear and attract some attention, then their relevance diminishes and users' interest shifts to fresher objects. Sorting the objects according to their current relevance is an inherently time-related task and we attempt to devise and compare several different algorithms to deal with it.

Finally, we use a news recommendation method and a testing framework proposed in (Zhou et al., 2009) to study the effect of uneven users' evaluating abilities and spamming activities on the recommendation process. The key question is whether high-quality objects survive the evaluation bias and reach their audience. We show that when the original recommendation model is generalized by introducing user reputation, spam objects can be entirely eradicated from the system by giving high enough weight to reputation. Even more, an intermediate weight of reputation is able to both effectively suppress spammers and improve recommendation performance.

While the devised artificial datasets and information filtering algorithms are simple, they are open to further generalizations and improvements. The resulting algorithms can be tested on large-scale real datasets, in the "living labs", and then they may eventually become part of the platforms QMedia and QScience.

Contents

Chapter 1

Introduction

Quality is the cornerstone of the QLectives project which aims to form Quality Collectives: self-organised decentralised communities. However, besides the quality of communities' members, it is also the quality of the created or shared content what is of great importance for the project. In the context of the project, quality is not a given and extrinsic attribute but rather a consensual outcome of evaluations or opinions given by the community's members. The task of information aggregation (or, in general, the management of information overload) is the focus of information filtering (Belkin, 2000; Hanani et al., 2001).

In this deliverable we focus on how to discover and promote high quality content in the ever-changing world where new potentially relevant objects constantly appear, lose their relevance (quality), and disappear. Breaking news of today become ordinary news tomorrow and unimportant news in one week. It is hence important to study algorithms that are not only able to find relevant objects but are able to do it soon, sooner than objects' relevance becomes obvious or even foregone.

To achieve these ends, we investigated three different systems. In Sec. 2.1, preferential attachment (users are driven towards popular objects) is mixed with heterogeneous abilities of recognizing the true quality of an object (good evaluators are driven more by the object's true quality than the object degree). As the network of user-

object links gradually grows, the goal is to recognize who are the best evaluators and which are the best objects. In Sec. 2.2, it is assumed that users are equal in their evaluation ability but the true qualities of objects are not constant but they rather decay with time (*e.g.*, a news which is relevant today and tomorrow but not in one month). In this highly dynamical setting where the network of user-object links grows and qualities decay, the goal is at any instant to be able to pick the objects with currently highest quality values. Finally, chapter 3 studies a recently proposed recommendation method (Zhou et al., 2009). The main goal is to discuss the effect of variable evaluation ability of users and the impact of spammers (users who intentionally submit low-quality objects to the system) on the method's performance (mostly via agent-based modelling).

Chapter 2

Algorithms for better quality rankings

Despite the extensive development of reputation and recommender systems (Resnick et al., 2000; Herlocker et al., 2004; Witten and Frank, 2005; Hastie et al., 2009), the role of time has been mostly neglected until now. Data mining algorithms usually focus on "who rated what and how" and ignore the time when the rating was given. However, considering time can bring new aspects which would not be possible otherwise: for example, it can help to distinguish an early "discoverer" of quality content from "followers" who merely join already existing trends. It was recently shown that time-including algorithms may be more resistant to spammers than standard ones (Noll et al., 2009). As pointed out in (Crane and Sornette, 2008), analysis of time evolution of object's popularity itself can also help to distinguish quality content from rubbish.

In this chapter we devise simple rules for artificial datasets corresponding to gradually growing bipartite user-object networks (such networks can represent views of videos by users on YouTube, reading of news on a newspaper's page, etc.). We assume that the choice of users is not random but partially driven by intrinsic quality of the objects. When mixed with the classical preferential attachment mechanism (Yule, 1925; Simon, 1955; Price, 1976; Barabási and Albert, 1999), this semi-random user selection may be set to produce data structurally similar to real data.

Faced only with the resulting user-object network, one may try to reconstruct the

true object ranking according to their intrinsic quality. Our goal is to find such ranking algorithms that are able to profit from the additional information provided by time and outperform simple time-free algorithms.

In Sec. 2.1 we study a case where users differ in their ability of recognizing high-quality objects and hence opinion of some of them is inherently more valuable than opinion of others. This is a standard situation and there are some standard algorithms are already known to fit it well (Kleinberg, 1999). Yet, we show that the inclusion of time may considerably improve the obtained results. Then in Sec. 2.2 we assume that relevance of all objects decays with time and new objects appear constantly. In result we obtain a highly dynamic system where objects appear and attract some attention, then their relevance diminishes and users' interest shifts to fresher objects. Sorting the objects according to their current relevance is an inherently time-related task and we attempt to devise and compare several different algorithms to deal with it. We conclude this chapter with a discussion of the achieved results and future research directions.

2.1 Users with heterogeneous evaluating abilities

As mentioned above, we aim to construct a realistic user-object bipartite network. Our reference is the Movielens data (see the web page of the GroupLens project at the University of Minnesota, `www.grouplens.org`) which comprises approximately 6 000 users, 4 000 objects, and 10^6 links between users and objects. In Fig. 2.1 we show the degree distribution for both objects and users in this real network. As it is common for many techno-social systems, this distribution is broad (though, not exactly a power law) (Newman, 2005).

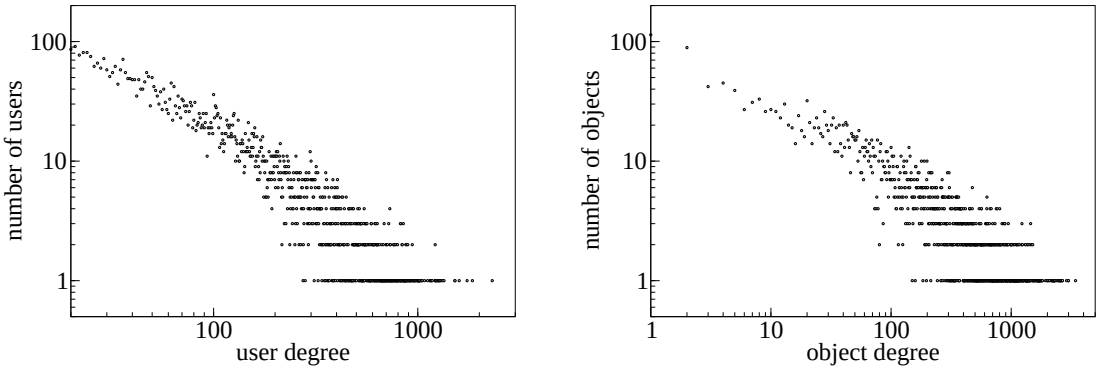

Figure 2.1: Degree distributions in the Movielens network.

2.1.1 Construction of artificial datasets

We assume that the number of users in the artificial data is fixed and we denote it as U. The number of objects grows with time and we denote it as $O(t)$ ($O_0 := O(0)$ is the initial number of objects). Note that time in our system is purely artificial—we increase it by one each time when a new user-object link is introduced to the system (has t is equal to the current number of links). The set of objects present in the system at time t we denote as $\mathcal{O}(t)$ (obviously, $O(t) = |\mathcal{O}(t)|$). For the sake of clarity we use Latin letters ($i, j, \ldots$) and Greek letters ($\alpha, \beta, \ldots$) for user- and object-related indexes, respectively. We denote the set of objects collected by user i as $\mathcal{C}_i$, the set of users who collected object α as $\mathcal{C}_\alpha$, the number of objects collected by user i (degree of user i) as k_i, and the number of users who have collected object α (degree of object α) as k_α.

We assume that the intrinsic quality of object α is $q_\alpha \in [0;1]$ and that the inherent rating ability of user i is $a_i \in [1; a_{\max}]$. In our simulations, we draw object quality values from the skewed distribution

$$f(q_\alpha) = (1+\beta)(1-q_\alpha)^\beta \tag{2.1}$$

where $q_\alpha \in [0;1]$ and $\beta \geq 0$ is a parameter. The distribution of quality is uniform for $\beta = 0$. The higher the value of β, the more abundant the low-quality objects are. The

distribution of user abilities is similar to Eq. (2.1), namely

$$g(a_i) = \frac{(a_{\max} - 1)^{1+b}}{1+b} (a_{\max} - a_i)^b \tag{2.2}$$

where $a_i \in [1; a_{\max}]$ and $b \geq 0$ is a parameter. Again, the higher the value of b, the more abundant the low-ability users are.

The network itself is constructed in a similar way as the seminal Barabási-Albert network (Barabási and Albert, 1999): starting with O_0 initial objects and all U users, at each time step we randomly choose one user and connect it to a semi-randomly selected object (see below). In addition, with a certain small probability P_N we introduce a new object to the system (its initial degree is, of course, zero). The time when object α was introduced we denote as T_α. The rate of adding new objects is chosen to reach the final number of objects, O.

The key element of the model is the selection rule describing how are users connected to objects. Similarly as in (Bianconi and Barabási, 2001), we use a generalized preferential attachment mechanism where the probability of selecting an object grows both with the object's degree and the object's quality (or, generally speaking, object's fitness). That means, when user i is to select an object, the probability of choosing object α is

$$P(i, \alpha) \sim q_\alpha^{a_i} (1 + k_\alpha)^{1/a_i}$$

where k_α is the degree of object α. This form ensures that users are attracted both to popular objects (due to the $(1 + k_\alpha)$ term) and to high-quality objects (due to the q_α term). The least able users ($a_i \approx 1$) are equally sensitive to quality and object degree while users with high ability ($a_i \gg 1$) are driven mainly by quality and much less by object degree.

Since we assume that the selected user must choose one yet uncollected object (*i.e.* each object cannot be collected by an individual user more than once), the correctly

normalized $P(i,\alpha)$ has the form

$$P(i,\alpha,t) = \frac{q_\alpha^{a_i}\left[1+k_\alpha(t)\right]^{1/a_i}}{\sum_{\beta\in\mathcal{O}(t)\setminus\mathcal{C}_i(t)} q_\beta^{a_i}\left[1+k_\beta(t)\right]^{1/a_i}} \tag{2.3}$$

where $\mathcal{C}_i(t)$ is the current set of objects collected by user i and hence $\mathcal{O}(t)\setminus\mathcal{C}_i$ is the set of the objects available at time t and yet uncollected by user i. Note that there is also a network model where a broad (scale-free) degree distribution can be obtained only using node fitness, without referring to preferential attachment (Caldarelli et al., 2002). However, object popularity has strong influence on user decisions in practice and therefore we believe that Eq. (2.3) is simple yet reasonable realistic way to model real user-object networks.

To test the described method for constructing artificial datasets, we compare basic properties of the resulting networks with features observed in real networks. To allow for a comparison with the aforementioned Movielens dataset, we set $U = 6\,000$, $O_0 = 1\,000$, and $O = 4\,000$. The resulting object degree distribution corresponding to $\beta = 0$ (uniformly distributed quality values), $b = 3$, and $a_{\max} = 80$, shown in Fig. 2.2, is close to a power law and it is also reasonably similar to the object degree distribution shown in Fig. 2.1. The user degree distribution is Gaussian due to the assumed random selection of users. While it would be simple to make it more realistic by assigning suitably distributed activity values to the users, we do not use this generalization because our focus is the ranking of objects and user degrees are hence of less relevance.

A further important dependency is that between object quality and degree. Fig. 2.3 shows that object degree and quality are positively correlated as expected. Furthermore, in Fig. 2.3a where $O_0 = O$ (*i.e.*, all objects are present from the beginning), the dependency between object quality and degree is rather narrow which implies that the object degree itself is a good ranking criterion. Hence, such a "static" system where new objects are not introduced constantly is in some sense so simple that there

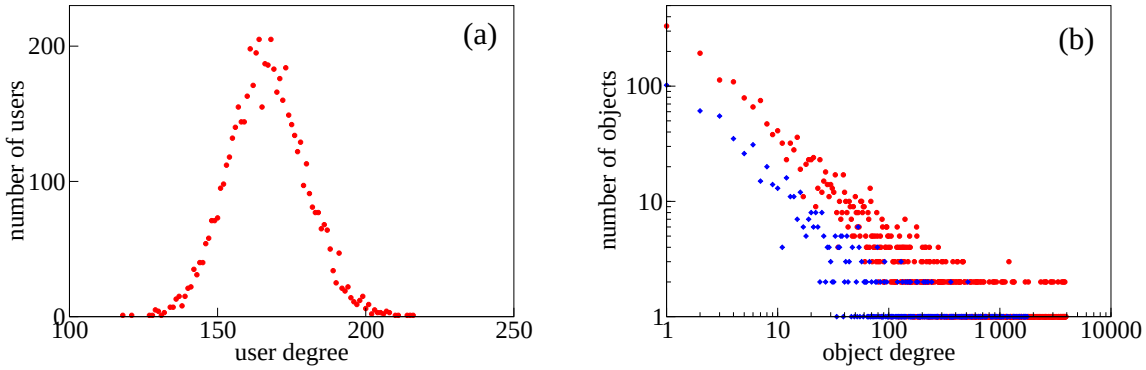

Figure 2.2: User (a) and object (b) degree distribution for $U = 6\,000$, $O = 4\,000$, $O_0 = 1\,000$, $\beta = 0$, $b = 3$, $a_{\max} = 80$, 10^6 ratings. (In (b), also object degree distribution for 10^5 is shown with blue diamonds.)

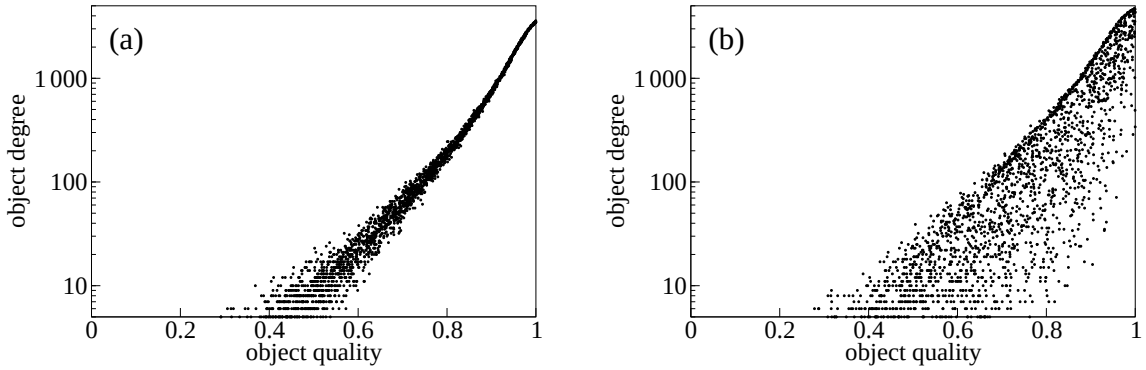

Figure 2.3: Object degree vs. object quality for different initial numbers of objects: $O_0 = 4\,000$ (a) and $O_0 = 1\,000$ (b) (other parameter values as in Fig. 2.2).

is no need for sophisticated ranking algorithms. On the other hand, in Fig. 2.3b where $O_0 = O/4$, there are many objects of near-perfect quality which, however, attract only a mediocre number of users. It's not surprising that these objects fail to succeed simply because they were introduced to the system too late. And it is exactly these objects on which we want to focus and allow them to rank high despite their late appearance and relatively small degree.

Finally, Fig. 2.4 shows the dependency of the average quality of objects collected by an individual user on the user's ability. Contrary to expectations, higher ability does not automatically mean that the use collects better objects. In particular, at the ability value approximately 3, the collection's quality is at its minimum. This behaviour is observed consistently for different values of the model parameters $O_0, \beta, a_{\max}, b$. To

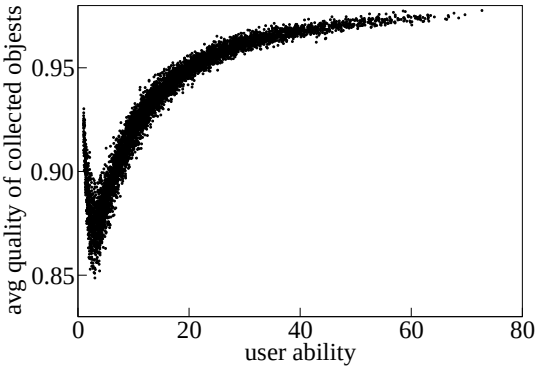

Figure 2.4: Average quality of a user's collection vs. user's ability (parameter values as in Fig. 2.2).

explain it, we can recall that the users with low ability are driven mainly by object degree and hence preferentially select popular objects which, as shown in Fig. 2.2, mostly have high quality. At the same time, users with intermediate ability values compromise between quality and degree with, as it turns out, poor results.

2.1.2 Algorithms

Once the artificial data is prepared, we may use a ranking algorithm to obtain estimated object quality values $\tilde{q}_\alpha$ and user ability values $\tilde{a}_i$ (equivalently, $\tilde{q}_\alpha$ and $\tilde{a}_i$ may already be the resulting rankings). The differences between obtained and true rankings are measured by Kendall's rank correlation coefficient τ which is defined as

$$\begin{aligned} \tau_U &= \frac{2}{U(U-1)} \sum_{i<j} \operatorname{sign}\left[(a_i - a_j)(\tilde{a}_i - \tilde{a}_j)\right], \\ \tau_O &= \frac{2}{O_c(O_c-1)} \sum_{\alpha<\beta} \operatorname{sign}\left[(q_\alpha - q_\beta)(\tilde{q}_\alpha - \tilde{q}_\beta)\right] \end{aligned} \tag{2.4}$$

where O_c is the number of objects collected by at least one user. Often it is the top of the ranking what interests us most. To reflect that, we add a second performance metric, success rate F_{20}, which is defined as the fraction of 20 highest quality objects that appear also in top 20 of the estimated ranking (hence $F_{20} \in [0;1]$). This measure is very different from τ which takes into account the whole ranking.

The benchmark object ranking algorithm is to simply rank the objects according to their degree. This popularity-based algorithm, POPULAR, has the obvious drawback of being unable to rank users because in the proposed model for creating artificial data, user's degree is independent of their ability. Originally developed to compute the importance of web pages, HITS algorithm distinguishes two different qualities of a web page: authorities are pages pointed to by many hyperlinks, hubs are pages pointing to many pages (Kleinberg, 1999). We implement this algorithm by assigning authority scores A to users and hub scores H to objects, both are initialized as $A_i^{(0)} = k_i$, $H_\alpha^{(0)} = k_\alpha$. The input data is represented by a $U \times O$ matrix D where $D_{i\alpha} = 1$ if user i has collected object α and $D_{i\alpha} = 0$ otherwise. The next iterations of scores are obtained as

$$A_i^{(n+1)} = \sum_\alpha D_{i\alpha} H_\alpha^{(n)}, \quad H_\alpha^{(n+1)} = \sum_i D_{i\alpha} A_i^{(n)}. \tag{2.5}$$

To avoid the divergence of the scores, normalization is applied after each iteration step such that $\sum_i A_i = 1$ and $\sum_\alpha H_\alpha = 1$. Iterations stop when the difference of consecutive score vectors, δ, is smaller than a certain threshold value Δ (in our simulations we compute δ as the average absolute difference of vector elements and use $\Delta = 10^{-8}$). Eqs. (2.5) represent mutually reinforcing scores: authorities pointing to by many hubs are strong authorities and hubs pointing to several highly rated authorities are popular hubs.

Motivated by the recent study of spamming-resistant algorithms, we generalize HITS by replacing the binary (zero or one) matrix elements $A_{i\alpha}$ with continuous-valued elements

$$A_{i\alpha}(t) = \begin{cases} 1 + Cx_\alpha & \text{if } \alpha \in C_i, \\ 0 & \text{if } \alpha \notin C_i, \end{cases} \tag{2.6}$$

where x_α is the fraction of time that object α didn't spend in the system, $x_\alpha = (t - T_\alpha)/t$, and C is a constant that we can optimize. The rest of the implementation is identical with original HITS. This new algorithm, HITS*, gives more importance to recent links. It is also possible to make values $A_{i\alpha}$ dependent not only on the age of the

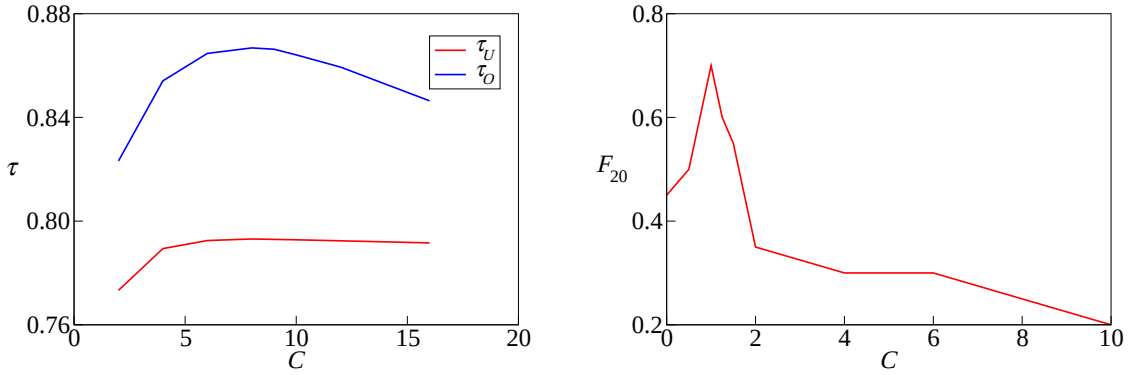

Figure 2.5: Optimization of HITS* with respect to τ (a) and F_{20} (b).

method	τ_O	τ_U	F_{20}
POPULAR	0.73	—	0.40
HITS	0.73	0.70	0.45
HITS* ($C = 1$)	0.79	0.75	0.70
HITS* ($C = 10$)	0.86	0.79	0.20

Table 2.1: Results of the respective algorithms.

object but rather on the age of the link. However, our tests suggest that performance of such a method is generally lower than that of HITS*.

Before comparing the described methods, HITS* needs to be optimized with respect to C. This optimization, shown in Fig. 2.5, yields the optimal value $C \approx 8$ with respect to τ_O and $C \approx 1$ with respect to F_{20} (which means that the optimization of HITS* is task-specific). Numeric tests further reveal that both HITS and HITS* converge fast and only a handful of iterations (typically ten) are needed to reach the specified convergence goal of $\Delta = 10^{-8}$.

Comparative tests of the three described algorithms are summarized in Tab. 2.1. They show that while the standard HITS algorithm scores only a mild improvement over the simple popularity ranking, herein proposed modification HITS* is able to significantly improve both τ_O and F_{20} (when tuned properly).

Albeit structurally similar to real datasets, the described artificial data has unrealistic temporal properties with degrees of most high-quality objects experiencing accelerated growth without limits. This behaviour is caused by the lack of a saturation factor

in the model—a drawback that we will remedy in the following section. It is now natural to expect that the acceleration of the degree growth, proportional to $k_\alpha/(t-T_\alpha)^2$, could be a meaningful criterion for ranking of the objects. This simple non-parametric quantity is indeed surprisingly successful and results in $\tau_O \approx 0.85$ which is almost as good as the results achieved with relatively sophisticated HITS* (on the other hand, the corresponding F_20 is 0.10). This good performance suggests that the proposed artificial data, based on preferential attachment mixed with heterogeneous user ability and object quality, is in fact rather simple and it would need to be made more complex to provide a challenging test bench for ranking algorithms.

2.2 Objects with time-dependent quality values

Here we aim to model a system where objects' relevance decays with time and hence it's important to select the relevant objects both accurately and rapidly. The artificial data is created similarly as in Sec. 2.1. Given a certain initial number of objects, O_0, we randomly choose one user in each time step and semi-randomly (assuming user's sensitivity to both objects' relevance and degree) select the object collected by the given user. A new object with zero links is introduced in each time step with a certain small probability.

Denoting the relevance of object α at time t as $R_\alpha(t)$, the probability that user i selects object α is assumed to be

$$P(i,\alpha) = \frac{R_\alpha(t)\left[1+k_\alpha(t)\right]}{\sum_{\beta\in\mathcal{O}(t)\setminus C_i(t)} R_\beta(t)\left[1+k_\beta(t)\right]}. \tag{2.7}$$

where the relevance values decay as

$$R_\alpha(t) = \frac{R_\alpha(0)}{1+(t/t_c)^{-d_\alpha}}. \tag{2.8}$$

As can be seen from Eq. (2.7), all users have identical ability, $\forall i:\ a_i = 1$. Heterogeneity

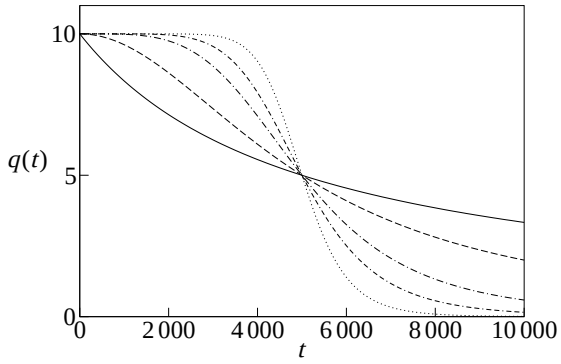

Figure 2.6: Relevance evolution for $R_\alpha(0) = 10$, $t_c = 5\,000$, and $d_\alpha = 1, 2, 4, 6, 10$.

is now assumed only for objects: they differ by their initial relevance values (some objects are little relevant already at the moment of their appearance) and also by their power-law exponents d_α which describe how fast the relevance values vanish. The characteristic lifetime of objects, t_c, would be easy to make different for each object but we do not study this generalization here. Fig. 2.6 shows typical shapes of the decay of relevance behaving according to Eq. (2.8).

In our simulations we construct networks containing either 10^5 or 10^6 user-object links. Values $q_\alpha(0)$ are drawn from the log-normal distribution with parameters $\mu = 1$ and $\sigma = 0.5$ and shifted by 1 (hence each object's initial quality is more than 1). A log-normal distribution of relevance values is used in order to obtain a few objects of very high relevance. Values of the exponents d_α are uniformly distributed in the range $[1; 10]$.

Fig. 2.7a shows the resultant degree distribution which is in both settings consistent with the one obtained from real data. Fig. 2.7b shows degree evolution of some objects and demonstrates that the obtained "ecology" is quite rich: objects growing rather slowly but steadily are mixed with "shooting stars" which grow only over a very limited period of time. Degree itself hence does not suffice to obtain a good ranking of objects and more sophisticated algorithms are necessary.

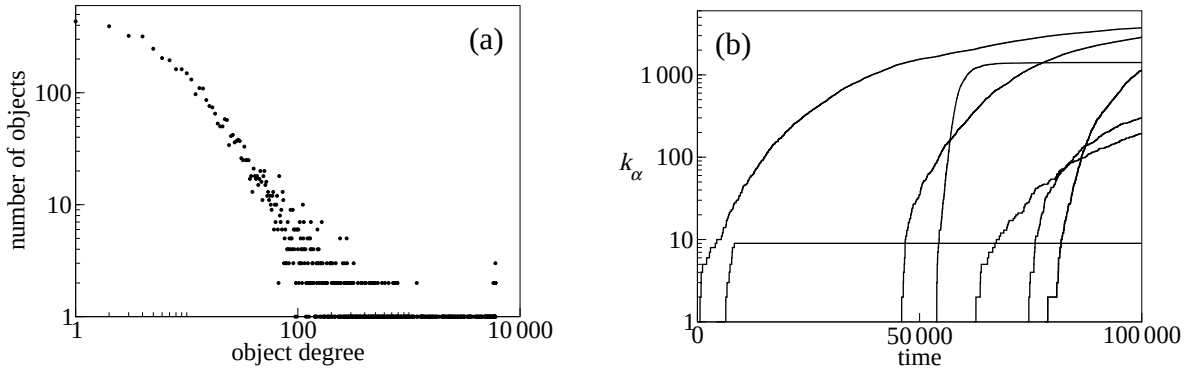

Figure 2.7: (a) Degree distribution for the artificial data with 10^6 links. (b) Degree evolution of some randomly selected objects from this data.

2.2.1 Algorithms

Our goal is to rank objects at a certain time t such that the obtained ranking agrees well with the current relevance values $R_\alpha(t)$. To this end, it is now not useful to use a variation of HITS because all users are now equal in their ability of selecting relevant objects and hence discriminating among them would not provide any substantial improvement. Due to the assumed homogeneity of users, all useful information is contained in the evolution of object degrees $k_\alpha(t)$. We denoted the time when user i collected object α as $t_{i,\alpha}$ and the time when object α has entered the system as T_α.

A simple ranking of objects can be obtained by ranking them according to the degree growth over a recent period Δt. That is, object scores are

$$S_\alpha(t) = k_\alpha(t) - k_\alpha(t - \Delta t) \tag{2.9}$$

where the time window length Δt is a parameter to be optimized. This simple algorithm, GROW, we use as a benchmark for more complicated approaches. This simple method is nowadays used in many popular news pages where users can find list of recently most popular articles (for users it is often possible to set the window length individually to best fit their own needs and preferences).

Another intuitive approach is to generalize the object degree such that old links are counted less. Assuming simple exponential suppression of old links (hence the

method's label EXP), object scores have the form

$$S_\alpha(t) = \sum_{i \in \mathcal{C}_\alpha} \mathrm{e}^{-(t-t_{i,\alpha})/\tau_c} \tag{2.10}$$

where t_c is a characteristic discounting time and $\mathcal{C}_\alpha$ is the set of users who have collected object α.

Finally, Eq. (2.7) mixes relevance with degree in such a way that relevance alone can be written as

$$R_\alpha(t) \sim \frac{P(i,\alpha)}{1+k_\alpha(t)}.$$

Since the number of links eventually linking to object α is proportional to the probability $P(i,\alpha)$, we can use the number of new links to approximate $P(i,\alpha)$. Thus the object scores can be computed as

$$S_\alpha(t) = \frac{k_\alpha(t) - k_\alpha(t-\Delta t)}{1+k(t-z\Delta t)} \tag{2.11}$$

where $z \in [0;1]$ is a parameter specifying whether we divide by the final object degree ($z = 0$), by the object degree at the beginning of the time window ($z = 1$), or we use some time in between. Since this method makes direct use of the preferential attachment formula (2.7), we label it as PREF.

To compare the suggested methods, we use the same performance metrics as in Sec. 2.1: Kendall's coefficient τ and the success rate F_{20}. However, both metrics need slight modifications. Firstly, the relevance of all objects decays with time such that after some time period they are not relevant and not collected any longer. To avoid the problem of ranking those inactive irrelevant object, we constrain the computation of τ to objects with the current relevance greater than 10^{-4}. Secondly, since relevance of objects decays fast, in the true ranking, top 20 objects often have very low degree and are hence hard to be discovered. Therefore we re-define F_{20} such that only objects with degree greater than $k_{\min}$ are included.

		$k_{\min} = 0$	$k_{\min} = 5$	$k_{\min} = 10$
method	τ		F_{20}	
GROW	0.402	0.042	0.117	0.209
EXP	0.550	0.079	0.178	0.280
PREF	0.449	0.296	0.683	0.774

Table 2.2: Average results of the described methods achieved on the artificial data.

Before the above described algorithms can be used, they need to be optimized with respect to their parameters Δt, τ_c, and z, respectively. The optimal values found by our numerical simulations (details not shown here) are $\Delta t = 5\,000$, $\tau_c = 1\,000$, and $z = 0$. Tab. 2.2 summarizes the average results achieved by individual algorithms in the time period $[20\,000; 100\,000]$ (the initial period is omitted from the averaging because the system is yet far from equilibrium there and results of all methods change significantly—see Fig. 2.8 for an illustration). PREF is best with respect to F_{20} but the rankings obtained with this algorithm are ambiguous when it comes to objects receiving no links during the past Δt turns (all those object receive score zero) and this lowers the resulting values of τ. On the other hand, EXP is best when τ is concerned but it fails considerably in F_{20}.

We can conclude that the artificial datasets constructed in this and the previous section reproduce several features observed in real datasets and allow for well controlled tests of information filtering algorithms. We proposed various simple algorithms and showed that they answer the given task (to rank the available objects) better than the standard methods.

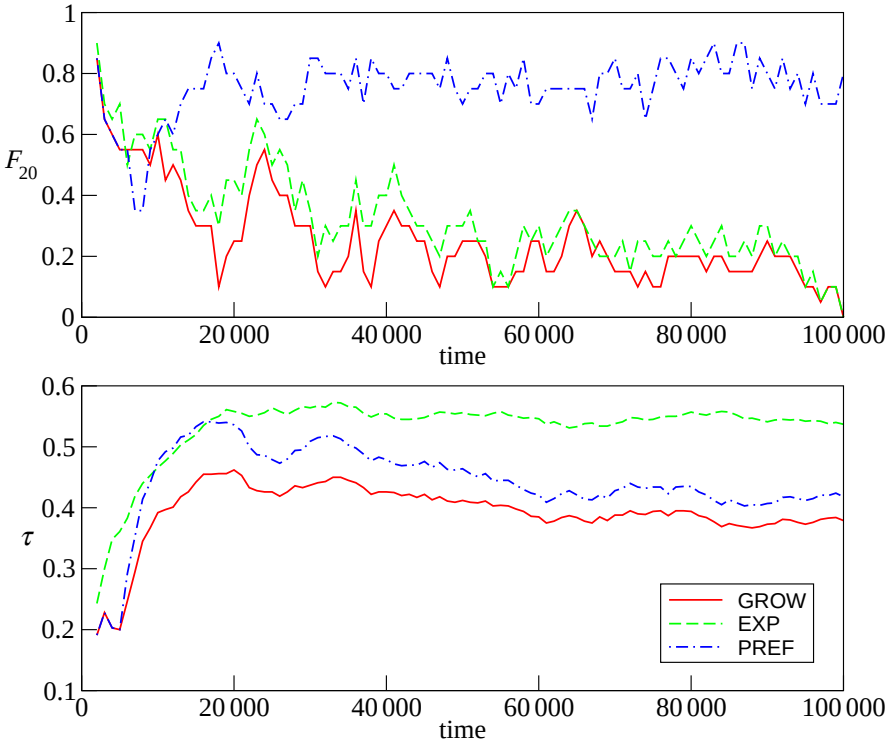

Figure 2.8: Resulting τ and F_{20} for respective methods at different stages of the data growth (time is equivalent to the number of links, F_{20} was computed with $k_{\min} = 10$).

Chapter 3

Evaluation errors and spamming in recommender systems

In (Zhou et al., 2009), they proposed an adaptive recommendation model which combines similarities in users' rating patterns with epidemic-like spreading of news on an evolving network. Since the model was studied using computer agent-based simulations, it provides an ideal test ground for further generalizations. In this chapter we focus on the effect of uneven users' evaluating abilities and spamming activities on the recommendation process. In both cases, the key question is whether high-quality objects survive the evaluation bias and reach their audience.

3.1 Recommendation model

Before proceeding to generalization, let's introduce the mentioned recommendation model and an agent-based framework for its tests. The studied system consists of U users (again labelled with Latin letters) and each user has S sources (when they approve an object, the recommendation score of this object is increased for the given user). This set of users and their sources can be interpreted as a directed network of users where node in-degree (the number of links pointing to a node) has a fixed value

S. Evaluation of object α by user i, $e_{i\alpha}$, can be either $+1$ (liked), -1 (disliked), or 0 (not evaluated yet). Recommendation score of object α for user i, $R_{i\alpha}$, is obtained on the basis of estimated similarity values s_{ij} between user i and this user's sources j. This similarity is estimated from the given evaluations: if i and j agree in evaluations of A_{ij} objects, while commonly evaluating N_{ij} objects, their similarity is obtained as

$$s_{ij} = \frac{A_{ij}}{N_{ij}}\left(1 - \frac{1}{\sqrt{N_{ij}}}\right). \tag{3.1}$$

The negative term $1/\sqrt{N_{ij}}$ serves to penalize user pairs with a small number of commonly evaluated objects (similarity estimates for these pairs are inevitably more biased and it is hence better not to rely on such pairs).

Recommendation of news is achieved on the basis of a spreading process which takes place on the directed user network. When user i introduces a new object α to the system, it is automatically assumed that the object was approved by this user. Object's recommendation score $R_{j\alpha}$ is set to s_{ij} for all users j who have user i as a source while this score is zero for all other users (news α is "spreading" from user i to this agent's followers). If object α is later approved by user j, its recommendation score increases by s_{jk} for all users k who have user j as their source, and so forth. For each user, available not rated objects are sorted according to their recommendation scores and top objects are recommended.

Since at the beginning we lack information about users, the initial assignment of sources to individual users is random. Starting from this random network configuration, as information gradually mounts, the user network is rewired at regular intervals. The two basic rewiring methods are

1. *Random local rewiring*: for each user, the source with smallest similarity value is replaced with a randomly selected user,

2. *Global rewiring*: for each user, similarity with all other users is computed and S most similar users are selected as sources.

While the random local rewiring is computationally cheap, it needs to be repeated many times to considerably improve the assignment of sources. On the other hand, the global rewiring makes the best use of the available information but it is computationally demanding.

For numerical tests of the described recommendation model, the following agent-based approach was introduced. User tastes and object attributes are represented by D-dimensional binary vectors. Taste vector of user i, $\boldsymbol{t}_i$, contains D_A ones ("active" tastes) and the remaining $D - D_A$ elements are zero. Attributes of each news is assumed to be identical with the user who has introduced a given news to the system. Users are chosen such that all possible different taste vectors are used—the number of users is hence $U = \binom{D}{D_A}$. In one step of the simulation, each user is active with certain probability p_A. If active, the user consumes and evaluates top three objects from his recommendation list and with small probability p_S submits a new object. The overlap of i's tastes and α's attributes is obtained using the scalar product

$$\Omega_{i,\alpha} = (\boldsymbol{t}_i, \boldsymbol{a}_\alpha). \tag{3.2}$$

When user i is asked to evaluate news α, she likes it if $\Omega_{i,\alpha} \geq \Delta$ and dislikes it otherwise.

Since we have full control of the system, it is possible to measure the differences of taste vectors between users and their sources. Since taste vectors are binary, it's enough to count the number of differing taste elements—its average value over all users and their sources, *average differences*, is the first measure of the system's performance. The second measure, *approval fraction*, is simply the relative fraction of approvals (likes) in the system. This measure is more practical than *average differences* and can be also used in real applications.

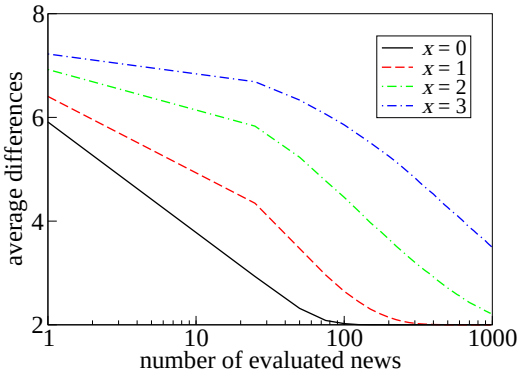

Figure 3.1: The average number of differences vs. number of evaluated news for various noise strengths ($x_i = x$ for all users). Simulation parameters: $D = 16$, $D_A = 6$, $U = 8\,008$, $S = 10$, $\Delta = 3$.

3.2 Effect of evaluation errors

Real users are not judging machines and errors are always present in their evaluations. To include this feature in simulations, we generalize Eq. (3.2) to the form

$$\Omega_{i,\alpha} = (\boldsymbol{t}_i, \boldsymbol{a}_\alpha) + u x_i \tag{3.3}$$

where u is a random variable uniformly distributed in the range $[-1;1]$ and $x_i > 0$ is the individual error strength of user i. Before testing the effect of errors on the complete object-spreading process described above, we study a simplified process where all users gradually evaluate all objects.

As shown in Fig. 3.1, evaluation errors have negative influence on the system's performance: the speed of convergence to the optimal state decreases with x. When $x \simeq \Delta$, the magnitude of errors becomes comparable with the signal intensity and the system needs an enormous number of evaluations to converge.

Now we study a heterogeneous set of users who differ by their level of activity (controlled by the probability of being active, p_A), their approval threshold (controlled by Δ), and their magnitude of errors (controlled by x). In particular, we set Δ_i uniformly distributed in the range $[2;4]$, x_i uniformly distributed in the range $[0;x]$, and probability of being active $p_i^A = 0.01/y$ where y is uniformly distributed in the range

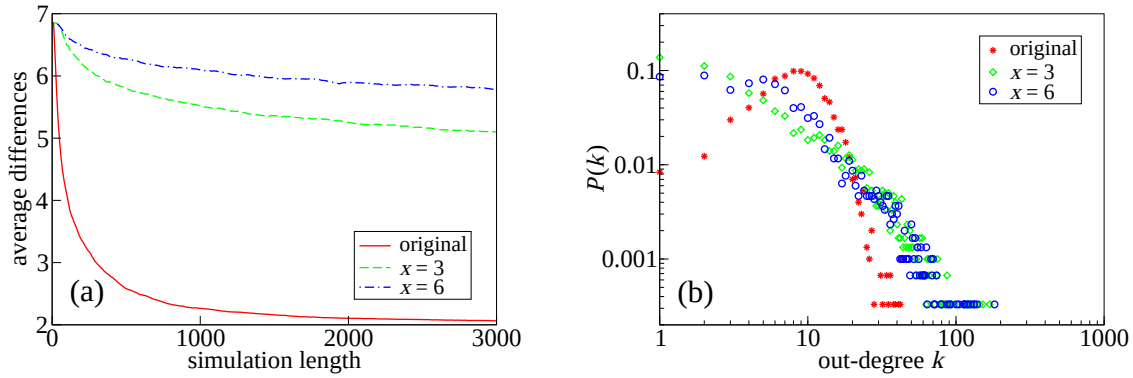

Figure 3.2: The evolution of average differences and the out-degree distribution for the original homogeneous setting and for two different heterogeneous settings. Simulation parameters: $D = 14$, $D_A = 6$, $U = \binom{14}{6} = 3003$, $S = 10$, global rewiring.

$[0.01; 1]$ (hence the probability ranges from 0.01 to 1 with low values significantly more often than the high ones). The probability of submitting a new object is set as $p_i^S = p_i^A/10$. This simple generalization makes the system more realistic and allows us to study the dependency between users' properties and their function in the system.

Since with erroneous users we observed very slow convergence to the optimal assignment of sources (see Fig. 3.1), with more sources of noise added (represented by variable levels of activity and demandingness) one can expect that convergence will become even slower. Fig. 3.2 compares the heterogeneous setting described above with the original homogeneous one where all users have identical activity ($p_A = 0.1$), submitting probability ($p_S = 0.01$), and approval threshold ($\Delta = 3$). We see that the convergence is indeed considerably slower in the heterogeneous setting and the optimal assignment of sources may even never be reached. At the same time, the out-degree distribution in heterogeneous cases is much different from the original homogeneous case as it exhibits a familiar broad shape. Notably, similar out-degree distributions are observed in many real systems (Newman, 2005). The studied simple model reproduces this feature when the amount of noise in the system is enough to prevent the rewiring process from driving the network to the optimal state. This suggests that real networks are perhaps also far from their best configurations.

It is now a natural question, how much the found properties of the directed user

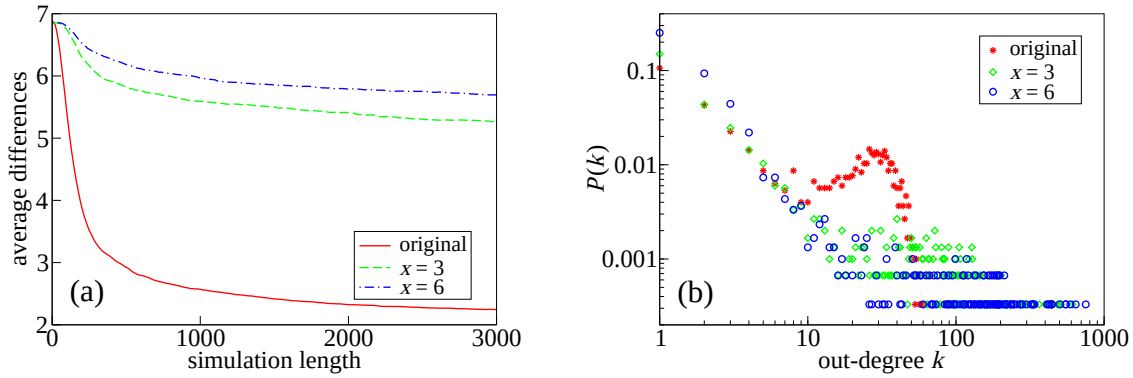

Figure 3.3: The evolution of average differences and the out-degree distribution for the original homogeneous setting and for two different heterogeneous settings. Parameter values as in Fig. 3.2, hybrid rewiring.

network depend on the chosen rewiring mechanism. To this end, we test a different mechanism which, similarly as the random local rewiring described in Sec. 3.1, is based only local information about the user's neighbourhood in the directed network. It is a hybrid mechanism where for user i, the least similar source is replaced by either the best among the sources of i's sources (with probability 90%) or by a random user (with probability 10%). Our tests show that this hybrid mechanism performs better than the individual methods which suffer from slow convergence (random selection of new sources) or convergence to a sub-optimal assignment (sources of sources selection). In fact, this hybrid procedure mimics the behaviour of real users who naturally attach themselves to friends of their friends but sometimes also experience an accidental encounter and make a new friend who was originally very distant in the social space. As shown in Fig. 3.3, the slow convergence to the optimal assignment of sources is present also for the hybrid rewiring. Further, the observed out-degree distribution is now even broader and it can be well approximated by a power-law with exponent around 1.6.

3.3 Robustness against spammers

A good recommender system should be able not only to promote high-quality content provided by gifted users but also to filter out bad content provided by malicious users (spammers). To test the response of the described system on the presence of spammers, we generalize Eq. (3.2) by assigning an additional quality factor Q_α to each news so that

$$\Omega_{i,\alpha} = Q_\alpha(\boldsymbol{t}_i, \boldsymbol{a}_\alpha) \tag{3.4}$$

Simple tests confirm that news with high quality are now more appreciated and popular than low-quality news. This means that the impact of spammers on the system is limited as their content does not have large audience. The reason for this behaviour is simple: when a low quality object is introduced to the system (regardless whether intentionally or not), it is sent to a small number of its introducer's followers who consequently disapprove it and effectively remove it from the system after harming only a handful of users. When quality values Q_α are drawn at random and do not depend on the user who submits the news, the speed of convergence as well as the approval fraction are qualitatively the same as in case without Q_α.

To further study the relation between quality of objects introduced by a user and this user's out-degree (*i.e.*, the number of users who are followers of this user), we assume that when a news is introduced, its quality Q_α is not chosen at random but rather it's copied from the user who introduces the news. Then good users consistently provide high-quality objects and malicious users consistently provide bad objects.

One could expect (Zhou et al., 2009) that since spammers tend to disagree with their followers (who hate their junk objects), they will soon lose these followers who attach themselves to better content providers. However, such disagreements are negligible compared with the total amount of user evaluations. In other words, if a spammer besides submitting low-quality content reasonably evaluates other objects, the such-created camouflage is sufficient to prevent spammer's followers from disconnecting.

We introduce *user reputation* to overcome this problem. The reputation score of user i is defined as

$$R_i = \left(1 - \frac{1}{\sqrt{|\mathcal{I}_i|}}\right) \sum_{\alpha \in \mathcal{I}_i} \frac{A_\alpha}{N_\alpha} \left(1 - \frac{1}{\sqrt{N_\alpha}}\right) \tag{3.5}$$

where $\mathcal{I}_i$ is the set of objects introduced by user i, A_α is the total number of approvals for object α and N_α is the total number of evaluations of object α. When $\mathcal{I}_i = \emptyset$, the reputation of user i is set to a small value ε. User reputation is then mixed with similarity in the rewiring process such that for user i, candidate neighbours j are sorted according to

$$m s_{ij} + (1 - m) R_j$$

where m is a mixing parameter. The extreme cases are $m = 0$ (selection of sources purely by their reputation) and $m = 1$ (selection purely by similarity which is identical with the criterion used up to now).

The system now consists of three distinct classes of users: average users who introduce objects with $Q = 1$, providers of good content ($Q = 2$), and spammers ($Q = 0$). The introduction of user reputation modifies the network evolution profoundly. When m is considerably smaller than 1, the average out-degree of spammers sinks and the average out-degree of high-quality users may reach considerable levels (see Tab. 3.1). On the other hand, if user reputation becomes too important ($m \gtrsim 1/2$), spammers are left without followers but recommendation diversity suffers (due to very high out-degrees of high-quality users), clustering coefficient of the network (Newman, 2003) increases, average number of differences increases, and the overall user satisfaction (measured by the approval fraction) decreases (see Fig. 3.4).

	average user degree		
	spammers	h-q users	clustering
$m = 1$	10.0	10.0	0.126
$m = 3/4$	7.4	10.4	0.136
$m = 2/3$	4.8	11.2	0.192
$m = 1/2$	0.0	31.2	0.301

Table 3.1: Average out-degree for spammers and h-q (high quality users) for different values of the mixing parameter. The out-degree for common users remains 10 (that is the fixed values of the in-degree) independently of m. Simulation parameters: $U = 8008$, $D = 16$, $D_A = 6$, $S = 10$, $\Delta = 3$, $p_A = 0.1$, $p_S = 0.01$.

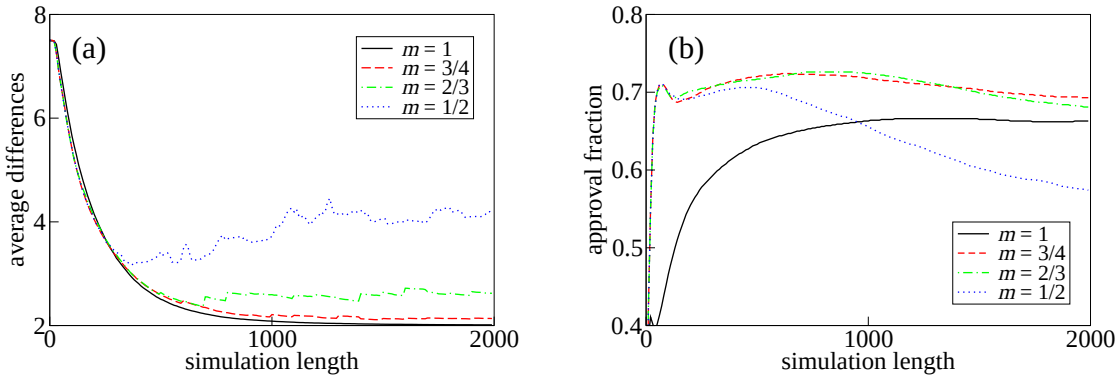

Figure 3.4: Average differences and approval fraction for different values of the mixing parameter m (parameter values as in Tab. 3.1).

Chapter 4

Summary and further questions

In this deliverable we aimed to study dynamic artificial systems where new objects constantly appear and to devise algorithms able of early detection and promotion of high-quality objects.

In Chapter 2 we addressed this task by examining two different systems. In the first system, all objects were continually collecting new links due to a mixture of preferential attachment and user sensitivity to object quality (we also assumed that users differ in their ability to recognize object quality). It is then natural that old objects have more links than new objects and hence any efficient algorithm must take age of objects into account. We showed that the proposed setting produces data reasonably similar with real datasets and that simple time-depending algorithms can outperform both simple ranking of objects according to their aggregate number of links as well as the standard HITS algorithm. In the second system, we assumed that object relevance decays with time and it is hence even more important to find the relevant objects as soon as possible. We again proposed several distinct ranking algorithms, tested them on the prepared artificial datasets, and identified strong and weak points of respective algorithms.

While results presented in Chapter 2 provide a sound test bench and suggest several simple yet efficient algorithms, there are several open research questions remain-

ing. Firstly, the artificial data can be improved to exhibit some other empirically observed features—degree correlations (which are known to be positive in most social systems and negative in most technological systems), clustering, and others. Secondly, as soon as the described algorithms are thoroughly tested and their dependency on the model parameters are understood, it is essential to test the algorithms on real datasets. While this provides a significant technical challenge (because detailed datasets including time are not simple to obtain), our study otherwise cannot be conclusive. Finally, the described time-related approach can be generalized from ranking algorithms to recommender systems where it may both pose new problems and yield new results.

In Chapter 3 we studied one recommender method and tested it using simple agent-based modelling. Our main goal was to investigate the influence of rating errors and spammers on the system. We found that considerable rating errors give rise to broad out-degree distribution of users which in some cases can be well approximated by a power law. This suggests that broad degree distributions observed in many real systems are perhaps partially due to significant errors and limited information available to the users. By tests assuming the presence of spammers (users providing low-quality content) we found that their impact on the system is limited. When the recommendation model is generalized by introducing user reputation, spam objects may be entirely eradicated from the system by giving high enough weight to reputation. While that weight of reputation is already damaging for the system, setting it to an intermediate weight is able to both suppress spammers and improve recommendation performance.

To summarize, we studied various systems where object quality plays a prominent role. We showed that simple rules are able to generate non-trivial artificial data on which devised algorithms can be tested in near-laboratory conditions. In all studied systems, exploitation of time patterns and user reputation contributed to finding and supporting high-quality objects. While the devised artificial datasets and information filtering algorithms are simple, they are open to further generalizations and

improvements. The resulting algorithms can be tested on large-scale real datasets, in the "living labs", and then they may eventually become part of the platforms QMedia and QScience.

Bibliography

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

Belkin, N. J. (2000). Helping people find what they don't know. *Communications of the ACM*, 43:58–61.

Bianconi, G. and Barabási, A.-L. (2001). Competition and multiscaling in evolving networks. *EPL*, 54:436–442.

Caldarelli, G., Capocci, A., Rios, P. D. L., and noz, M. A. M. (2002). Scale-free networks from varying vertex intrinsic fitness. *Physical Review Letters*, 89:258702.

Crane, R. and Sornette, D. (2008). Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 105(41):15649–15653.

Hanani, U., Shapira, B., and Shoval, P. (2001). Information filtering: overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11:203–259.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, Berlin, 2 edition.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. 22(1):5–53.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256.

Newman, M. E. J. (2005). Power laws, Pareto distributions and Zipfs law. *Contemporary Physics*, 46:323–351.

Noll, M. G., Yeung, C. A., Gibbins, N., Meinel, C., and Shadbolt, N. (2009). Telling experts from spammers: Expertise ranking in folksonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 612–619. ACM.

Price, D. J. D. S. (1976). A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society or Information Science*, 27:292–306.

Resnick, P., Zeckhauser, R., Friedman, E., and Kuwabara, K. (2000). Reputation systems. *Communications of the ACM*, 43(12):45–48.

Simon, H. A. (1955). On a class of skew distribution functions. *Biometrika*, 42:425–440.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edition.

Yule, G. U. (1925). A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London B*, 213:21–87.

Zhou, T., Medo, M., and Zhang, Y.-C. (2009). Adaptive model for recommendation of news. *EPL*, 88:38005.