Karlsruhe Institute of Technology

Incentive Mechanisms and Quality Assurance for Peer Production

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Conny Kühne

aus Köthen (Anhalt)

Tag der mündlichen Prüfung: 03.02.2016
Erster Gutachter: Prof. Dr.-Ing. Klemens Böhm
Zweiter Gutachter: Prof. Dr. Rudi Studer

Acknowledgements

First and foremost, I would like to thank Prof. Klemens Böhm, my supervisor for this dissertation, for his great help and support during my years as a doctoral student in Karlsruhe. His tireless hunting for unsound arguments, hard-to-understand sentences, and his pursuit and the resulting extinction of the dreaded passive voice made the quality of my papers considerably better. His constructive feedback greatly improved my research.

I would also like to thank my second advisor Prof. Rudi Studer for his willingness to undertake the ungratifying task of reading and commenting on my thesis.

Next, I am very thankful to my colleague and good friend Christian von der Weth. Not only did I greatly enjoy his companionship during our hikes through the Black Forest, countless hours in the university gym, and night games of IPD hallway tennis. I also am much indebted for his suggestions and proofreading for this dissertation.

Further, I am grateful to the students I supervised. They greatly helped with coding, and came up with valuable research ideas. In particular, I would like to thank Christian Hütter and Benjamin Lautenschlager. They contributed large portions of the Consensus Builder platform and greatly helped conducting the experiments with it.

My thanks also go to the members of our working group for making my time at the IPD a great and fun experience. In particular, I thank Stephan Schosser, Guido Sautter, Björn-Oliver Hartmann, and Andranik Khachatryan for valuable discussions *about* as well as *after* work.

Last, but definitely not least, I would like to thank Sabine Tittel for enduring my moods and ensuring that somehow this dissertation got finished.

Zusammenfassung

Die kollaborative Erstellung von Wissensartefakten hat sich zu einem weit verbreiteten Phänomen entwickelt. Wikipedia und Linux sind nur zwei bekannte Beispiele für Wissensartefakte, die von virtuellen Gemeinschaften autonomer, gleichberechtigter Mitwirkender (Peers) erstellt werden. Der dabei verwendete Produktionsmodus wird auch als Peer Production[1] bezeichnet. Es handelt sich dabei um einen dezentralisierten Prozess, der ohne eine hierarchische Kontrollinstanz auskommt. Das Peer-Production-Modell vereint die Vorteile der Skalierbarkeit, da sich einzelne Peers leicht hinzufügen lassen, mit der der Robustheit, da sich einzelne Peers leicht ersetzen lassen.

Wenn eine Gemeinschaft von Gleichberechtigten ohne das Eingreifen einer zentralen, übergeordneten Instanz Wissen erstellt und pflegt, stellen sich zwei wichtige Fragen: (1) Wie lassen sich Peers zur Mitarbeit motivieren? Die Qualität vieler virtueller Gemeinschaften leidet unter einer zu geringen Beteiligung. Insbesondere für kleine Gemeinschaften ist dies ein Problem. Diese benötigen, im Gegensatz zu sehr großen Gemeinschaften wie Wikipedia, eine verhältnismäßig hohe Anzahl von Beiträgen pro Mitglied, um die kritische Masse für ihr Fortbestehen zu erreichen. (2) Wie lässt sich die Qualität des erstellten Wissens feststellen und sichern? Die Abwesenheit einer koordinierenden Kontrollinstanz darf die Datenqualität nicht negativ beeinflussen.

Beiträge und Vorgehen

Der Hauptbeitrag dieser Arbeit ist die Beantwortung der zwei oben genannten Fragen innerhalb unterschiedlicher Einsatzgebiete von Peer Production, nämlich, (i) der kollaborativen Erstellung von *strukturiertem* Wissen, (ii) der Genauigkeit von Klassifikationsverfahren basierend auf Bewertungen und (iii) der Bewertung wissenschaftlicher Peer-Reviews durch Autoren.

Die Kollaborative Erstellung Strukturierten Wissens

Unser Hauptfokus liegt auf der kollaborativen Erstellung strukturierten Wissens, am konkreten Beispiel von Ontologien. Wir untersuchen dabei das folgende Szenario. Eine virtuelle Gemeinschaft von Gleichberechtigten erstellt eine Ontologie, welche eine bestimmte Anwendungsdomäne konzeptualisiert. Zur Qualitätssicherung bewerten die Mitglieder die Beiträge der anderen. Um die einzelnen Mitglieder der Gemeinschaft zur Mitarbeit zu motivieren, belohnen wir sie mit Punkten. Die Anzahl der Punkte, die ein Mitglied erhält, richtet sich nach der Qualität und der Quantität der erstellten

[1] Zu deutsch etwa 'Fertigung durch Gleichberechtigte'.

Beiträge. Dabei berechnen wir die Qualität eines Beitrags anhand der Bewertungen, die er bekommen hat. Da Bewertungen eine zentrale Rolle spielen, müssen auch sie von hoher Qualität sein. Ein viel versprechender Ansatz, um qualitativ hochwertige Bewertungen zu erhalten, sind so genannte *Mechanismen für ehrliche Bewertungen* (engl. *honest rating mechanisms* – HRMs). HRMs belohnen subjektive Ehrlichkeit in Situationen, in denen kein objektives Wahrheitskriterium vorhanden ist. HRMs wurden allerdings bisher kaum empirischen Tests unterzogen, insbesondere nicht im Kontext der kollaborativen Erstellung von Wissen.

Ausgehend vom oben skizzierten Szenario beantworten wir unter anderem folgende Forschungsfragen zur Erstellung strukturierten Wissens: (i) Wie beeinflussen bewertungsabhängige Belohnungen die Qualität und die Quantität von Beiträgen? (ii) Bewirkt ein HRM eine höhere Bewertungsqualität als z.B. eine feste Belohnung pro Bewertung?

Um diese Fragen zu beantworten, haben wir eine Anwendung zur dezentralen, anreizbasierten Erstellung strukturierten Wissens namens *Consensus Builder* entwickelt. Consensus Builder verfügt über feingranulare Bewertungsmechanismen, insbesondere über einen HRM. Basierend auf den Forschungsfragen formulieren wir eine Anzahl von Hypothesen und testen diese ausgiebig in einer Reihe von kontrollierten Feldexperimenten. Im Gegensatz zu Laborexperimenten, sind die Bedingungen unserer Experimente nah an Realwelt-Bedingungen und verfügen damit über eine hohe externe Validität. Im Gegensatz zu einfachen Beobachtungsstudien erlauben uns die Experimente, Kausalzusammenhänge durch die Variation verschiedener Parameter und den Vergleich ihrer Auswirkungen nachzuweisen. Die Experimente zeigen, dass Bewertungen ein zuverlässiges Maß für die Qualität von Beiträgen innerhalb einer Gemeinschaft zur Erstellung strukturierten Wissens sind. Der Einsatz von Bewertungen und von bewertungsabhängigen Belohnungen führt zu einer Steigerung der Qualität der Wissensbasis. Zudem können wir feststellen, dass ein HRM die Qualität von Bewertungen in der Mehrheit der Experimente signifikant steigert. Allgemein zeigen die Studien, dass Consensus Builder sich sehr gut zur Erstellung strukturierten Wissens eignet. Damit leisten unsere Ergebnisse einen wichtigen Beitrag zur bedeutenden allgemeinen Forschungsfrage, wie man qualitativ hochwertiges strukturiertes Wissen im großen Stil erstellen kann.

Genauigkeit von Bewertungsbasierten Klassifikationsverfahren

Viele virtuelle Gemeinschaften müssen die von ihnen erstellten Beiträge klassifizieren. Das heißt, sie müssen jeden Beitrag einer bestimmten Klasse aus einer vorgegebenen Menge von Klassen zuordnen. Als Beispiel stelle man sich wieder eine Gemeinschaft zur Erstellung von Ontologien vor. Hier müssen die Mitglieder etwa entscheiden, ob ein gegebenes Element der Ontologie eine Klasse oder eine Instanz ist, oder ob der Name eines Begriffs korrekt oder inkorrekt ist. Die Klassifikation wird dabei durch ein bewertungsbasiertes Klassifikationsverfahren ausgeführt, z.B. durch einen einfachen Mehrheitsentscheid. Als Eingabe für die Klassifikation bewerten die Mitglieder, ebenso wie im Szenario zur Erstellung strukturierten Wissens, die Beiträge anderer. Die Bewertungen entsprechen dabei den möglichen Klassen (im obigen Beispiel etwa 'Klasse/Instanz' oder 'korrekt/inkorrekt'). Zur Schätzung der Klasse von Beiträgen und zur Schätzung der Fehlerrate von Bewer-

tern haben Dawid und Skene einen Algorithmus vorgeschlagen. Diesen bezeichnen wir im Folgenden als Dawid-Skene-Algorithmus (DSA). Es gibt zahlreiche Vorschläge in der Literatur, DSA und auf DSA basierte Algorithmen in Crowdsourcing-Umgebungen einzusetzen.

Trotz seiner Beliebtheit hat DSA zwei Schwachstellen: (1) Seine Genauigkeit bricht ein in Umgebungen mit einer hohen Fehlerrate der Bewerter. Diese können auftreten, wenn das Thema der virtuellen Gemeinschaft inhärent schwierig ist. Ebenso können sie entstehen, wenn die Gemeinschaft durchsetzt ist von einer großen Anzahl von Spammern, bösartigen Bewertern oder Bewertern, die in ihrem Urteil voreingenommen sind und deshalb verzerrte Bewertungen abgeben. (2) DSA ist anfällig gegen Kollusionsattacken. In einer Kollusionsattacke koordinieren sich einzelne Bewerter, um bestimmte Datenobjekte identisch zu bewerten. Eine Kollusionsattacke ist von Vorteil für Bewerter, wenn diese abhängig von ihrer geschätzten Fehlerrate belohnt werden. Da DSA den Bewertungen der Angreifer unfair hohe Gewichte zuordnet, können Kollusionsattacken die Klassifikationsgenauigkeit von DSA extrem verringern.

Wir schlagen *Goldstrategien* vor, um die Genauigkeit von DSA in Umgebungen mit hoher Fehlerrate zu erhöhen und um Kollusionsattacken abzuwehren. Goldstrategien machen Gebrauch von Goldobjekten, d.h. Beiträgen, deren wahrer Wert DSA bekannt ist. Wir gewinnen Goldobjekte, indem wir ausgewählte Beiträge von vertrauenswürdigen Experten bewerten lassen. Allerdings führt das in der Literatur übliche zufällige Auswählen von Beiträgen als Goldobjekte nur zu einer unbefriedigenden Erhöhung der Klassifikationsgenauigkeit. Statt dessen nutzen unsere Goldstrategien den Grad der Übereinstimmung der Bewertungen eines Beitrags als Selektionskriterium. Da Goldobjekte Kosten verursachen ist es unser Ziel, ihre Anzahl so zu wählen, dass der Nettonutzen der Goldobjekte, d.h. ihr Nutzen abzüglich ihrer Kosten, maximiert wird. Diese Anzahl an Goldobjekten a priori zu bestimmen ist praktisch unmöglich. Wir schlagen einen adaptiven Algorithmus vor, der sukzessive die Anzahl der Goldobjekte erhöht und automatisch entscheidet, wann keine weiteren Goldobjekte mehr benötigt werden.

Unser Vorgehen ist wie folgt: Zunächst betrachten wir die Eigenschaften von bewertungsbasierten Klassifikationsverfahren, wie die des einfachen und die des gewichteten Mehrheitsentscheids. Anschließend evaluieren wir verschiedene Goldstrategien für DSA mit Hilfe von Simulationsexperimenten. Wir demonstrieren, dass bestimmte Goldstrategien die Performanz von DSA in Umgebungen mit hoher Fehlerrate stark zu steigern vermögen. Ferner zeigen wir, dass Goldstrategien ein wirksames Mittel sind, um Kollusionsattacken gegen DSA zu verhindern: Einerseits begrenzen sie den Schaden der Kollusionen bezüglich der Klassifikationsgenauigkeit. Andererseits vermindern sie den Vorteil, den die Angreifer aus der Kollusionsattacke ziehen können und machen somit die Attacke unattraktiv. Schließlich zeigen wir, dass der adaptive Algorithmus die optimale Anzahl von Goldobjekten für jede Kombination aus Goldstrategie und Simulationsszenario mit hoher Genauigkeit findet.

Die Bewertung der Bewerter – Autoren Bewerten Wissenschaftliche Gutachten

Die wissenschaftliche Gemeinschaft ist der wahrscheinlich am längsten währende Einsatzbereich für Peer Production. Zur Qualitätssicherung benutzen die meisten wissenschaftlichen Disziplinen das Verfahren der Begutachtungen durch Gleichgestellte: Peer Review. Das Verfassen eines Gutachtens für einen wissenschaftlichen Artikel erfordert einen hohen zeitlichen und kognitiven Aufwand seitens des Gutachters. Die Anreize für das Verfassen von qualitativ hochwertigen Gutachten sind vergleichsweise niedrig. Das liegt vor allem daran, dass Gutachter in der Regel unentgeltlich arbeiten und zudem meist anonym bleiben. Obwohl die meisten Gutachten durchaus von hoher Qualität sind, gibt es auch einen nicht zu vernachlässigenden Anteil von Gutachten minderer Qualität. Dies zeigen z.B. die zahlreichen Diskussionen über die Vor- und Nachteile des Peer-Review-Systems in den Fachpublikationen verschiedener Disziplinen.

Wir meinen, dass die Autorenbewertungen von Gutachten potentiell den Peer-Review-Prozess verbessern können. Sie könnten etwa dazu dienen, qualitativ hochwertige Gutachten zu identifizieren und die entsprechenden Gutachter dafür zu belohnen. Jedoch sind die Kriterien für eine adäquate bewertungsbasierte Belohnung nicht offensichtlich. Zum Beispiel ist anzunehmen, dass die Entscheidung über die Annahme bzw. Ablehnung eines Artikels zur Publikation einen entscheidenden Einfluss auf die Bewertung durch die Autoren hat. Daher sind direkt auf Autorenbewertungen basierende Belohnungen für Gutachter höchstwahrscheinlich nicht objektiv.

Um empirische Erkenntnisse über die Wahrnehmung von Autoren zu Gutachten zu gewinnen, haben wir eine Studie mit den Autoren einer Informatik Fachkonferenz durchgeführt. Das Hauptziel der Studie war die Identifizierung von Kriterien zur Erkennung qualitativ hochwertiger Gutachten. Um dies zu erreichen, haben wir Bewertungen für Gutachten in den Peer-Review-Prozess der Konferenz integriert. Autoren konnten dazu die Gutachten, die sie für ihre Artikel erhalten hatten, anhand einer Vielzahl von Kriterien bewerten. Die Kriterien beinhalteten unter anderem die Nützlichkeit der Gutachterkommentare für die weitere Arbeit der Autoren oder den Aufwand, den der Gutachter investiert zu haben schien. Außerdem konnten Autoren direkt die Noten ('Originalität', 'Technischer Beitrag' etc.) bewerten, die ihnen die Gutachter gegeben hatten. Auf Basis der Studie haben wir eine umfangreiche Analyse durchgeführt.

Die Ergebnisse der Studie sind wie folgt: Interessanterweise bewerten Autoren die Noten der Gutachten überwiegend als adäquat (zur Verfügung standen 'zu niedrig', 'adäquat', 'zu hoch'). Wenig überraschend favorisieren Autoren Gutachten, die ihnen gute Noten zuteilen. Dabei beeinflussen die Einzelnoten der Gutachter die Bewertungen der Autoren jedoch in unterschiedlichem Ausmaß. Ferner bewerten Autoren Gutachten dann gut, wenn sie diese als hilfreich für ihre weitere Arbeit einschätzen, wenn sie die Kommentare gerechtfertigt finden und wenn sie den Eindruck haben, dass sich der Gutachter Mühe beim Verfassen des Reviews gegeben hat. Interessanterweise bleiben diese Zusammenhänge auch dann bestehen, wenn man den Einfluss der Gesamtnote des Gutachtens statistisch ausschließt. Die Entscheidung über Annahme bzw. Ablehnung eines Artikels sowie die Selbsteinschätzung der Kompetenz des Gutachters haben nur einen geringen Einfluss auf die Bewertungen der Autoren. Ausgehend von diesen Ergebnissen diskutieren wir

mögliche Ansätze zur Berechnung von Belohnungen für Gutachten.

Fazit

In dieser Dissertation haben wir die Fragen der Qualitätssicherung und der Motivationssteigerung in Peer Production untersucht. Wir konnten mit Hilfe von umfangreichen Feldexperimenten die Wirksamkeit der anreizbasierten Bewertungsmechanismen für die Erstellung strukturierten Wissens nachweisen. Unsere eigens entwickelte Anwendung Consensus Builder erwies sich in den empirischen Studien als sehr gut geeignet zur Erstellung strukturierten Wissens durch virtuelle Gemeinschaften. Außerdem haben wir auf Bewertungen basierende Goldstrategien entwickelt, mit deren Hilfe sich die Klassifikationsgenauigkeit des Dawid-Skene-Algorithmus' verbessern lässt. Dadurch lässt sich die Qualität der Klassifikation von Beiträgen in Peer-Production-Umgebungen wesentlich steigern, die durch einen hohen Anteil von Spammern und voreingenommenen Peers gekennzeichnet sind. Die den Nettonutzen maximierende Anzahl von Goldobjekten pro Strategie wird dabei mit hoher Genauigkeit durch den von uns vorgeschlagenen adaptiven Algorithmus gefunden. Des Weiteren bieten die Goldstrategien einen wirksamen Schutz gegen Kollusionsattacken. Schließlich haben wir umfangreiche empirische Ergebnisse darüber gewonnen, wie Autoren die Qualität von Gutachten im wissenschaftlichen Peer-Review-Prozess bewerten. Diese Ergebnisse haben wir genutzt, um mögliche Ansätze zur Berechnung von Belohnungen für Gutachten zu diskutieren. Einer dieser Ansätze vermag es, den verzerrenden Einfluss der Gutachternoten auf die Bewertungen größtenteils aufzuheben. Dies könnte in Zukunft einen Beitrag leisten, die Qualität von wissenschaftlichem Peer Review zu verbessern. Zusammenfassend zeigt diese Dissertation, wie der Einsatz von Bewertungsmechanismen und darauf aufbauenden Anreizmechanismen sowohl die Qualität als auch die Quantität von Beiträgen in Peer-Production-Umgebungen maßgeblich steigern kann.

Contents

1. Introduction

> Peer (noun): a person of the same age, status, or ability as another specified person.
>
> *(Oxford English Dictionary)*

The collaborative creation of knowledge artifacts has become a ubiquitous phenomenon. Wikipedia and Linux are perhaps the best-known examples for knowledge artifacts created collaboratively by a huge number of peers. Further well-known examples include the news aggregator Slashdot, the movie database IMDB, the question and answer forum Stackoverflow, the open learning community Peer to Peer University, and the digitization of cultural works by Project Gutenberg. These projects use a mode of production known as peer production [Ben02, Ben06]. Peer production refers to a decentralized production process where individual contributors work on a common project without a hierarchical organization. The peer-production model has the advantages that it is both scalable when adding further users and robust because individual users can be replaced.

However, when a community of peers, instead of an organization with a coordinating authority, creates and maintains knowledge artifacts, two questions arise: (1) How to motivate the peer-production workers? For example, many online communities suffer from under-contribution [BLW+04], i.e., only a minority contributes. In many peer-production communities, the number of contributions per member is Power-law distributed [Wil08, PHT09, MMM+11, Gil13]. In other words, a minority of 'power users' does the lion's share of the work – the vast majority of users free rides. This is bothersome in particular for communities with few members, e.g., online communities within corporations. Because of their small size, these communities need a high proportion of active contributors to reach a critical mass [BPSW10, RMJ10, SW14]. In contrast to large communities, they cannot easily tolerate a large proportion of free riders. (2) How to ensure and assess quality? The absence of a coordinator must not compromise the quality of the created knowledge.

In this dissertation, we investigate these questions within different settings of peer production:

(i) **Peer Production of Structured Knowledge.** Our main focus is the collaborative creation of *structured* knowledge, e.g., in the form of ontologies. We study how peer ratings can ensure the quality of the knowledge created: peers review each others' contributions and rate them according to the quality they perceive. Further, we analyze how incentive mechanisms based on the peer ratings can motivate the peer-production workers in this setting.

(ii) **Accuracy of Classification Schemes for Peer-Rating Online Communities.** We investigate a setting, where a community uses the peer ratings to classify contributions (in addition to using them for quality assurance and assessment). In particular, we study how to increase the classification accuracy in the presence of low-competence raters.

(iii) **Reviewing the Reviewers – Author Perception of Scientific Peer Reviews.** We investigate a mechanism for quality assurance in the sciences: peer review. Specifically, we analyze how authors rate the peer reviews they have received, and how the authors' ratings, in turn, can be used to increase the quality of the reviews.

In the following, we motivate the specific research questions of each setting in greater detail. At the same time, we discuss the contributions of this dissertation.

1.1. Peer Production of Structured Knowledge

We envision the following scenario. An online community collaboratively creates structured knowledge for a given domain. (We use the term *structured knowledge* as referring to any kind of interrelated, conceptualized information, ranging from a set of terms with informal relationships, like a hierarchy of tags, to a fully axiomatized ontology; cf. [NCA08].) For example, think of a project whose participants are geographically distributed, and who all contribute to a common knowledge base. Members of the online community review contributions created by the other members and rate the contributions according to the quality perceived. To motivate the individual members to contribute, we reward them with points corresponding to the quantity and the quality of their contributions. We compute the quality of a contribution based on the ratings it has received. Members can later convert the gathered points into external rewards. This could be gift coupons as with Epinions, or system privileges as with Slashdot.

Since we use ratings to compute the quality of contributions, the ratings themselves have to be of high quality as well. To gain high-quality ratings, *honest rating mechanisms* (HRMs) have been proposed [Pre04, MRZ05, JF06]. An HRM rewards subjective truthfulness in settings where no objective truth criterion is available. However, empirical studies on HRMs are rare. In particular, HRMs have not been studied in the context of collaborative knowledge creation, to the best of our knowledge.

Based on the above scenario, we study the following research questions regarding the creation of structured knowledge.

1. Are ratings a reliable measure of the quality of contributions?

2. How do rating-dependent remunerations for contributions affect the quality and the quantity of contributions?

3. Does an HRM induce ratings of higher quality compared to a fixed reward per rating?

To answer these questions, we have developed a platform for the collaborative creation of structured knowledge called *Consensus Builder*. It features fine-grained rating and incentive mechanisms, in particular an HRM. Based on the research questions above, we formulate a number of hypotheses and test them in a series of controlled field experiments in six different online communities. We control the experiments to be able to gain insights into causal effects of the tested mechanisms. Observational studies cannot achieve this. The experimental setups are close to the real-world scenario envisioned. In particular, participants in the experiments have used Consensus Builder from home or from their workplace to create structured knowledge.

1.1.1. Contributions

We make the following contributions regarding the peer production of structured knowledge:

- We have developed a platform for the collaborative creation of structured knowledge. It provides functionality to browse the knowledge base and to add or manipulate data items. Further, it features the rating-based incentive mechanisms discussed above.
- We discuss how mechanisms that reward the quantity and the quality of contributions, as well as the quality of ratings, can be applied to the creation of structured knowledge.
- To answer our research questions above, we formulate five hypotheses and we design controlled experiments to test them.
- Based on our platform, we have conducted extensive empirical studies to test the reward mechanisms w.r.t. the creation of structured knowledge. In these studies, we test our hypotheses in a variety of settings that are close to real-world environments, and with participants of different backgrounds.
- We show that the usage of rating mechanisms, and the usage of fully rating-dependent rewards for good contributions, increase the quality of contributions. Further, we show that an honest rating mechanism improves the quality of ratings.

1.2. On the Accuracy of Classification Schemes for Contributions in Peer-Rating Online Communities

Many online communities must classify their contributions, i.e., decide to which class of a set of predefined classes a contribution belongs. As an example, consider an online community that creates an ontology. In addition to contributing entries to the ontology, the community must classify the entries. That is, it must decide, e.g., if a given entry is a class or an instance, or if a name of an item within the ontology is correct or not. As input for the decision, the members of the community rate each others' contributions.

Ratings in this scenario correspond to the possible classes of an entry. E.g., ratings could have values "class/instance" if the community must decide if an item is a class or an instance in the ontology. Or, "correct/incorrect" if the community must decide about the correctness of an entry. After the ratings have been submitted, the community classifies the contributions by aggregating the contributions' ratings.

One simple scheme for rating aggregation is the well-known majority vote, i.e., the decision for the class that receives the majority of the ratings in its favor. Despite its simplicity, majority vote can achieve a decently high accuracy but only if the quality of ratings is sufficiently high. Aggregating ratings by means of weighted majority vote can increase the classification accuracy compared to majority vote, provided that weighted majority vote knows the individual competence of each rater, i.e., his/her probability of rating correctly. Intuitively, weighted majority vote assigns a higher weight to high-competence raters than to low-competence raters. Yet, rater competencies are unknown in general.

For this case of unknown competencies, Dawid and Skene proposed an expectation-maximization algorithm [DS79] to estimate the competencies of the raters and to classify the contributions accordingly. In the following, we refer to this algorithm as the Dawid-Skene algorithm (DSA). DSA has originally been developed to combine opinions of multiple physicians for medical diagnosis. With over 450 citations, DSA is one of the most widely-cited algorithms for classifying items based on ratings by raters with unknown competencies. In recent years, there have been a lot of proposals to use DSA – and algorithms based on or closely related to DSA – in particular for crowdsourcing settings [WRW+09, WIP11, RY12, WIP].

However, DSA has two major shortcomings: (1) It performs rather poorly if the mean competence of the community is low, in particular if it is less than random. Such a low mean competence can occur if the topic of the community is inherently difficult, or if the community is afflicted by a large fraction of spammers, malicious or biased raters. (2) It is vulnerable to collusion attacks. In a collusion attack, raters coordinate to rate the same data objects with the same value. A collusion is beneficial for raters if they are remunerated based on the estimated quality of their ratings. It allows colluders to artificially increase their remuneration while saving the cognitive effort for determining the truthful values of the contributions they rate. Further, since DSA assigns an inflated weight to the ratings of colluders, a collusion can severely damage the accuracy of DSA.

To overcome these two shortcomings, we propose *gold strategies*. Gold strategies utilize the notion of *gold objects*, i.e., contributions that DSA knows the true class of. Knowing the true class of a contribution with certainty allows DSA to estimate more accurately the competence of the raters who gave ratings to this contribution. This in turn, increases the accuracy of the estimates for non-gold objects. The existing approach simply selects gold objects randomly. However, this does not suffice to increase the accuracy of DSA much. In contrast to the existing approach, the gold strategies we propose select contributions based on the ratings the contributions have received. Specifically, they select contributions based on the level of agreement between raters, i.e., to what extent members of the community agree on the class of a given contribution. Trusted experts rate the selected contributions which turns these contributions into gold objects. However, trusted experts

and thus gold objects are costly. Consequently, our goal is to use the number of gold objects that maximizes the *net benefit*, i.e., the benefit of gold objects minus their costs. Determining that number a priori is infeasible. We propose an algorithm that adaptively determines the number of gold objects based on runtime information.

1.2.1. Contributions

We make the following contributions for classification schemes in peer-production settings:

- We analyze properties of majority vote, weighted majority vote, and DSA under varying assumptions, for example w.r.t. the competence distribution of the raters, or the distribution of the ratings in different settings.
- A common approach in the literature to differentiate between high- and low-quality raters is to evaluate their ratings by means of a set of randomly selected gold objects. In contrast, we propose gold strategies (i.e., selecting contributions based on specific criteria for evaluation by expert raters) that use the level of agreement between the ratings of the community as a selection criterion. We evaluate the effectiveness of the gold strategies in various settings by means of simulation. We show that, in low-competence settings, selecting contributions based on the level of agreement is vastly superior to the existing approaches that use randomly selected gold objects.
- We propose an adaptive algorithm that determines the number of gold objects in order to maximize the net benefit. Instead of fixing a predetermined number of gold objects, the adaptive algorithm adds gold objects iteratively and automatically decides when to stop adding further gold objects based on runtime information. We show that the adaptive algorithm determines the optimal number of gold objects for each gold strategy and each setting with high accuracy.
- We study the effects of collusion attacks against DSA. To the best of our knowledge, we are the first to do so. We show that gold strategies based on the level of agreement are highly effective for countering collusion attacks.

Our results are somewhat orthogonal to DSA. In principle, they are applicable to methods related to DSA [IPW10, WRW+09, RY12] as well.

1.3. Reviewing the Reviewers – Author Perception of Scientific Peer Reviews

Science is one of the most long-standing models of peer production [Hay09]. Most scientific communities use peer review as their de facto standard for quality assurance. Reviewing a scientific paper includes grasping its content, deciding on appropriate scores, and formulating valuable comments. This requires considerable intellectual effort and time. However, the incentive to write high-quality reviews tends to be somewhat low. A

possible reason for this is that reviewers remain anonymous to the authors and to the scientific community. Most reviewers do provide high-quality reviews. At the same time, the ratio of reviews of lower quality is non-negligible, at least according to the perception of authors. Various discussions regarding the pros and cons of peer reviewing in various scientific communities show this [CKG02, Nat06a, BBC+07].

We believe that feedback given by authors has potential to improve the review process. In particular, we deem review ratings a promising means for identifying high-quality reviews and for remunerating reviewers.[1] However, the specifics of a remuneration based on review ratings are not obvious. For instance, we must assume that accept/reject decisions influence the perception of authors. Therefore, rewarding reviews directly based on the ratings they receive from authors is unlikely to be objective. To illustrate: A review of a rejected paper is likely to obtain low ratings. Had the same paper been accepted, on the other hand, the review would presumably receive higher ratings, should that assumption hold.

To gain empirical insights into authors' perception of reviews, we have conducted a study with authors of a peer reviewed computer science conference. The main objective of the study was to determine which criteria are potentially useful to identify high-quality reviews and thus to determine an adequate basis for reviewer remuneration. To this end, we incorporated review ratings into the review process. Authors could assess each review they had received according to a broad selection of criteria, such as helpfulness of review comments, or the perceived effort of the reviewer to understand the paper.

1.3.1. Contributions

We make the following contributions regarding the peer production of peer reviews:

- We carry out a detailed analysis of author perception of peer reviews. Among others, we address the following questions: How strongly do the characteristics of the review, in particular the review scores, as well as the accept/reject decision, affect author ratings? Which of the different aspects of the review influence the authors' perception of the overall review quality most?

- We investigate how to remunerate peer reviewers based on author ratings. To our knowledge, we are the first to do so. We discuss a suitable metric to remunerate reviewers that neutralizes possible effects of the review process, e.g., the effects of the accept/reject decision, on author ratings.

1.4. Outline

This dissertation is structured as follows: We start by discussing the preliminaries that are required to understand the following investigations in Chapter 2. The preliminaries include a discussion of motivations for peer production in general and for motivation in

[1] The remuneration could, for example, be in the form of 'best reviewer' awards.

online communities in particular. Further, we discuss honest rating mechanisms as well as structured knowledge, ontologies in particular.

Next, we study the peer production of structured knowledge in Chapter 3. This includes the presentation of our tool Consensus Builder, as well as the discussion of our experiments, their results, and the lessons learned.

In Chapter 4, we investigate classification schemes for contributions in peer-rating online communities. Starting with a discussion of the properties of different classification schemes, we propose gold strategies. We evaluate the effectiveness of gold strategies for increasing the classification accuracy in low-competence communities, and for countering collusion attacks.

Chapter 5 introduces our study on author perception of peer reviews in computer science. It discusses the setup and the results of the study. Utilizing the study results, we discuss a remuneration function for reviewers based on author ratings.

Chapter 6 concludes.

1.5. Bibliographic Note

Portions of this dissertation have been published in [HKB08, KB14] (collaborative creation of structured knowledge), in [KB06, KB15] (classification schemes for contributions in peer-rating online communities), and in [KBY10] (author perception on peer review).

2. Preliminaries

This chapter introduces topics that are necessary to understand the rest of this work. This includes (i) a discussion of under-contribution and motivation in peer-production settings, (ii) an introduction to honest rating mechanisms, and (iii) a discussion of structured knowledge, in particular of ontologies.

2.1. Under-Contribution and Motivation in Peer Production

We discuss the literature on under-contribution and motivation in peer production. This includes an introduction to the various psychological models for explaining motivation. In particular, we analyze the literature to answer the following questions:

- How pervasive is under-contribution in peer-production communities?
- What are the different types of motivation and how do they affect peer production?
- What motivates people to contribute to peer-production artifacts, in particular in open-source software development, and in online communities?

2.1.1. Under-Contribution

Many peer-production communities suffer from under-contribution (also called under-provision), that is, only a minority contributes. The vast majority of users, on the other hands, free rides [BLW+04, Wil08, PHT09, MMM+11, PCL+07, Gil13]. For example, in February 2012, Wikipedia had 476 million unique visitors but only 85,163 active editors with more than five edits [Zac12]. In other words, about 0.02% of the users where active contributors. Moreover, the number of contributions per active participant in most peer-production communities follows a Power-law distribution: a tiny fraction of very active 'power users' creates the vast majority of contributions. For instance, among Wikipedia editors, the top 0.1% contribute nearly half of the value (44%) as measured in number of words viewed by users [PCL+07]. Similarly, 10% of the top contributors to open-source software account for 72% of the total codebase [GP00].

According to Kraut and Resnick, under-contribution can be a problem even in highly successful communities, like Wikipedia. For example, a quality assessment conducted by Wikipedia in 2010 found that out of roughly 900,000 articles evaluated on the English Wikipedia, two-thirds were stubs, i.e., articles "containing only a few sentences of text which is too short to provide encyclopedic coverage of a subject" [KR12]. Wilkinson and Huberman [WH07] find that article edits on Wikipedia follow a log-normal distribution. That means that a small number of articles that cover highly relevant or visible topics

attract a disproportionately high number of edit operations whereas a large number of articles receive very little edits. Similarly, Gilbert finds "widespread under-provision" [Gil13] of votes on the social-news site Reddit. Roughly half of all valuable content is overlooked on its first submission to the site.

Under-contribution is even more damaging to smaller communities. To reach a critical mass [BPSW10, RMJ10, SW14], i.e., to be sustainable in the long run, small communities need a higher proportion of active contributors and/or a higher number of contributions per member than large communities.

Models to Explain Under-Contribution

Several theoretical models try to explain why under-contribution is so wide-spread in peer production. For example, economic theory predicts that voluntary contributions in peer production will be under-provided because they are public goods [Ols65]. That is, once they are available, everyone can benefit from them without incurring costs. Since peers contribute voluntarily, they gain no monetary benefit from contributing. This makes free riding, i.e., consuming without contributing, the rational choice. So, there arises a social dilemma, where everyone would be better of if everyone contributed something, but there is a strong temptation to free ride on others contributions [KKRK12]. However, not everyone free rides all the time. In fact, as experience and many experiments show, many people contribute to public goods on a regular basis [Cap13].

Another theoretical model to explain under-contribution is *social loafing*. Social loafing refers to the tendency of individuals to expend less effort when working in a group than when working individually. For example, social loafers (also called lurkers in the context of online communities) follow the content of online communities, but choose not to contribute. Karau and Williams [KW93] integrate various theories about social loafing into their *collective effort model* to explain what motivates people to contribute to groups. Like most economic models, the collective effort model assumes that individuals try to maximize the expected utility of their actions. The model assumes that individuals will contribute to a collective task only to some extent. The exact extend depends on how much they expect their effort to translate into a valuable outcome. The motivation of an individual to contribute to a collective task depends on how strongly he/she perceives the relationship (a) between individual performance and group performance, (b) between group performance and group outcomes, and (c) between group outcomes and individual outcomes.

Because under-contribution is such a widespread phenomenon, there has been a great interest in investigating motivation for peer production in general, and user motivation in online communities in particular, as we will see in the following.

2.1.2. Types of Motivation

In general, the psychological literature, as well as the literature on motivation in peer production, distinguishes between two broad categories of motivation – intrinsic and extrinsic. Intrinsic motivation is "based in the innate, organismic needs for competence

and self-determination" [DR85, p. 32]. Intrinsic motivation makes an activity interesting and enjoyable. Further, it makes an activity likely to be performed for its own sake as opposed to as a means to an end [DR85]. In contrast, extrinsic motivation refers to the performance of an activity in order to achieve an outcome. Extrinsic motivation stems from an external source. This could be rewards like money, status, or grades, or other external regulations, like threats of punishment [DR85, KR12]. For example, some people might slay monsters in the multi-player online game World of Warcraft because they enjoy the task itself (intrinsic motivation). Others might do so because they are motivated by the status gained (extrinsic motivation) from achieving a high level in the game [KR12].

Some authors further distinguish between strictly intrinsic motivation, strictly extrinsic motivation, and internalized extrinsic motivation [DR00]. The latter is realized when people assimilate formerly extrinsic motives such that they become personally endorsed values that are integrated into the sense of self [DR00]. For example, people might start exercising to attain some outcome like fitness or weight loss, but might later internalize the importance of exercising as their own value and thus exercise more volitionally.

We note that the distinction between intrinsic and extrinsic motivation is not undisputed. For example, Reiss [Rei04] questions the distinction. Instead, he identifies 16 basic desires or motives such as power (desire to influence), curiosity (desire for knowledge), status (desire for social standing and attention), the desire for social contact, or the desire to eat. His theory states that the satiation of each of the basic desires produces an intrinsic feeling of joy. Further, he emphasizes individual differences in the expression of these desires. For example, some people have more desire for social contact than others.

Intrinsic and Extrinsic Motivation for Peer Production

Kraut and Resnick [KR12] discuss how to use both types of motivation to encourage participation in online communities: Intrinsic motivation in online communities and other peer-production settings can be increased for example by combining contribution with social interaction. For open-source software projects, this can be achieved, for example, by holding conferences. In online communities, (virtual) social contact is often provided in form of discussion forums, comment features, or other social features of the respective community platform. Further, community designers can enhance intrinsic motivation by providing clear goals and feedback, and challenges that are adjusted to people's skills. Online communities enhance extrinsic motivation by means of reputation mechanisms, system privileges, as well as tangible rewards like gift certificates or money. We discuss particular examples of extrinsic motivation further below.

Does Extrinsic Motivation Undermine Intrinsic Motivation?

Psychologist disagree on the relationship between the two types of motivation. Some argue that intrinsic and extrinsic motivation both enhance each other. For example, optimal work performance occurs when jobs are interesting and challenging and employees are rewarded for their work [PL68, CBP01]. Others state that extrinsic rewards undermine

(or crowd out) intrinsic motivation [Dec71]. For example, Lepper et al. [LGN73] assigned children that liked drawing to three experimental groups and let them draw pictures. In a first session, the first group received an expected (previously announced) reward for the drawings, the second group received an unexpected reward, and the third group received no reward at all. In a second drawing session, now unrewarded for all, children of the first group chose to draw less than children of the other two groups.

Further, different meta-analyses come to different conclusions whether tangible external rewards crowd out intrinsic motivation or not (see for example [PM13] for an overview). [CP02, CBP01] try to resolve the controversy. They argue that extrinsic rewards increase intrinsic motivation and performance on tasks that are of low initial interest. For high interest tasks, rewards have positive effects on motivation when they are intangible (e.g., in the form of verbal praise) or unexpected. Expected, tangible rewards do not affect high initial intrinsic motivation negatively when they are given for achieving predefined performance levels (e.g., scores) or when they are given for exceeding peers. On the other hand, tangible rewards given independently of performance, e.g., for a number of completed units, can decrease intrinsic motivation. Kraut and Resnick [KR12] discuss these findings w.r.t. online communities. They formulate the design claim that "adding a task-contingent reward (for doing or finishing a task, regardless of performance) to an already interesting task will cause people to be less interested in the task and to perform it less often. The effect will be larger for monetary rewards than for prizes, status rewards, and charitable donations".

Does the crowding-out effect matter in peer-production settings? The crowding-out effect of extrinsic on intrinsic motivation seems limited in existing peer-production settings. Moreover, external rewards seem to have rather positive effects in most peer-production settings. For instance, Lakhani et al. as well as Roberts et al. investigate motivations to contribute to open-source software projects. Contrary to their expectation, they find no evidence of extrinsic motivations crowding out intrinsic motivations [RHS06, LW05]. In many online communities, external rewards in form of reputation, status, or karma points seem to be an essential ingredient to stimulate contributions [MMM$^+$11, LR04]. For example, Farzan et al. [FDM$^+$08] could substantially increase contributions in an online community employed within a corporation by introducing an incentive mechanism based on reputation points. Harper et al. study question-and-answer sites and find that paying more money for an answer leads to longer, better answers [HRRK08]. Thus, even though under certain conditions, extrinsic rewards might undermine intrinsic motivation, in general, extrinsic rewards seem to have no negative effects in peer-production settings. Finally, even if extrinsic motivation does crowd out intrinsic motivation, designers of online communities are usually much more concerned with the overall combined effect of extrinsic and intrinsic motivation [KR12]. In that sense, the crowding-out effect does not seem to matter much.

2.1.3. Motivation to Contribute to Open-Source Software

Lakhani and Wolf [LW05] find that open-source software contributors are motivated by a combination of various intrinsic and extrinsic factors. As mentioned above, they find that neither of the investigated factors dominates or crowds out the others. The most important intrinsic factors are a sense of creativity and the feeling of an obligation/connection towards the group. About 40 percent of the open-source contributors are paid by the respective employers to participate.

Similarly, Roberts et al. [RHS06] find that contributors to open-source software projects have multiple motivations. They differentiate between strictly extrinsic motivation in form of payment, internalized extrinsic motivation (contributing to solve a problem of personal use or benefit, or to enhance status and career opportunities), and strictly intrinsic motivation (activities that satisfy the need for competence, control, autonomy, and enjoyment). They find no evidence of extrinsic motivations crowding out strictly intrinsic motivations. Further, they find that monetary compensation and status motivation lead to above-average participation levels, while strictly intrinsic motivation does not. They speculate that intrinsic motivation may not be associated with better performance because aspects that make an activity interesting might come at the expense of attention towards the overall outcome. For example, contributors might work at perfecting a minor feature while loosing sight of the overall goal.

2.1.4. Motivation in Online Communities

Wikipedia

Nov studies the question "What motivates Wikipedians?" per questionnaire [Nov07]. His main findings are that contributors to Wikipedia report to be primarily motivated by fun (example questionnaire item: "Writing/editing in Wikipedia is fun.") and ideology (example questionnaire item: "I think information should be free."). Interestingly, while fun is moderately correlated with contribution level, ideology is not. The author offers the explanation that opinions about ideology might be strong but do not translate into actual behavior, exhibiting a case of "talk is cheap". Similarly, Yang and Lai [YL10] find that intrinsic motivation and internal concept-based motivation ("I consider myself a self-motivated person who likes to share knowledge") have the greatest impact on contributions to Wikipedia, while extrinsic motivation (status within the community and respect from others) plays only a minor role.

Kittur et al. [KPK09] investigate the conflict between two different kinds of motivation for peer production. On the one hand, the authors argue that users self-select tasks that "scratch their personal itch". On the other hand, peer-production communities often require a significant amount of maintenance work that individual members do not find rewarding. Their hypothesis, why this kind of group work still gets done, is that the strong identification of individual peers with their subgroups influences these peers to do more group-related work. They test the hypothesis based on log files of *WikiProjects*. A WikiProject is a group of editors who team up to improve Wikipedia usually focusing on a particular domain. Upon joining a WikiProject, editors are more likely to work on

project-related content. Further, editors shift their contributions towards coordination and maintenance work rather than production work. This suggests, that group influence can play an important role even in presumably self-directed peer-production systems.

Panciera et al. argue that Wikipedians, more specifically, 'power editors' who made at least 250 edits, "are born not made" [PHT09]. That is, they differ systematically from the less active majority of editors in the following categories: their initial activity level is higher and stays higher than that of other editors, they invoke community norms more often, and they produce edits of higher quality, and thus can be considered the "essential core" of the community. In contrast, Solomon and Walsh [SW14] investigate WikiProjects and find that the participation of power users in early stages of the projects is less valuable to sustainability than the collective contributions of less active majority of editors.

Finally, Halfaker et al. investigate reasons for the steady decline of the number of Wikipedia editors since 2007 [HGMR13]. They show that the usage of algorithmic tools for quality control that automatically reject contributions is responsible for the decline, since it acts as a demotivator that is driving away newcomers. Further, they find that Wikipedia's formal mechanisms for defining group norms and rules have calcified against changes, in particular those changes proposed by newcomers.

Online Communities Using Extrinsic Motivation

Other communities rely more strongly on extrinsic forms of reward than Wikipedia. For example, the technology related news-aggregator Slashdot ("News for nerds, stuff that matters.") rewards users with *karma* points [LR04]. Users can post news stories and related comments. To avoid information overload and to filter bad comments, Slashdot employs a distributed moderation system. Moderators, i.e., users above a certain karma threshold, can rate the quality of comments with scores from -1 to +5. Users receive karma points for different activities such as moderating comments or posting comments that receive high scores. To remove bad moderators, Slashdot employs a meta-moderation system where meta-moderators can rate ratings as either fair or unfair. Thus, karma points stimulate extrinsic motivation both by the privileges they grant and because they are a status signal to the community. A similar system is employed by other social-news sites such as Reddit [Gil13] or Hacker News [Hac14].

The question & answer site Stack Overflow [MMM$^+$11] relies heavily on a moderation system as well: participants can vote questions and answers of others up and down and suggest to close inappropriate questions. To motivate users, Stack Overflow utilizes several "highly effective" [MMM$^+$11] extrinsic factors: The points gathered for upvoted comments and questions serve as reputation and are also converted to system privileges. Further, the site awards badges for participation such as 'Enthusiast' ('Visited the site each day for 30 consecutive days.') or 'Autobiographer' ('Completed all user profile fields') [Sta14]. Moreover, public profiles demonstrate a user's expert knowledge to the community of peers or potential employers.

Social Science Theories for Motivation in Online Communities

The GroupLens research group [Gro14] built their own community for movie recommendations called *MovieLens* [CLA+03]. Within this community, they have conducted various studies. To investigate how to motivate users, they have primarily deployed theories from social science.

For example, Beenen et al. [BLW+04] utilize the collective effort model (see above) as well as goal-setting theory [LL02] to stimulate contributions to the MovieLens community. They test a number of hypotheses that are based on predictions from the theoretical models. They find that users contribute more when they are reminded of the uniqueness of their contributions and when they are given specific and challenging goals. However, other predictions from the social science theories did not materialize. For example, individual goals ("rate 8 movies") were not more motivating than group goals ("rate 80 movies in a group of ten"). Further, contrary to what the collective effort model predicts, MovieLens users will not rate more movies when (i) the personal benefit they receive from doing so or (ii) the benefit they provide to the community is made salient.

Similarly, Rashid et al. [RLT+06] utilize the collective effort model to fix the problem of under-contribution in MovieLens. They find that displaying messages in the user interface reminding users of the unique value of their contributions increases contribution quantity.

Cosley et al. [CFK+05] also deploy the collective effort model to study user motivation. Specifically, they investigate how a peer's contributions are affected by being overseen by other peers. To test this, they let users of the MovieLens community check data entered by other users via online forms. They find that being overseen has no affect on initial quality, i.e., on the quality of data when first entered by the users. However, the final quality, i.e., the quality of the movie data after peers have checked it, was improved by the checking. They find no difference between peer and expert oversight.

Lampe et al. [LWVO10] find that users continue to participate in an online community for different reasons than those that led them to the site originally. Additionally, they find that a sense of belonging was important to registered as well as to anonymous users.

Under-Contribution of Ratings

Ratings are a form of contribution. Therefore, they tend to suffer from under-contribution as well. For example, [Gil13] finds widespread under-provision of ratings on Reddit. Despite sharing the characteristic of under-contribution with contributions in general, there also seem to exist unique intrinsic motivations for contributing ratings. For example, ratings in online forums usually follow a bimodal, u-shaped distribution, with most of the ratings being either very good or very bad [HPZ06, AP00]. Hu et al. [HPZ06] propose a 'brag-and-moan' model to explain the distributions. The model assumes that consumers only choose to rate when they are either very satisfied with the purchased products (brag), or very dissatisfied (moan).

2.1.5. Discussion

The works discussed above can provide design recommendations to increase motivation within online communities. However, even though there exists a large body of literature on motivation in peer production, the problem of under-contribution remains [Gil13]. Further, we are not aware of publications that experimentally test extrinsic incentives to rate the quality of contributions. Moreover, motivation to contribute might differ between successful communities such as Wikipedia or fun-based communities like MovieLens on the one hand, and communities for the creation of structured knowledge on the other hand. In Section 3.1.2 we discuss related work directly concerned with motivation for the collaborative creation of structured knowledge.

2.2. Honest Rating Mechanisms

The idea behind honest rating mechanisms (HRMs) is to elicit honest ratings in the absence of an objective truth criterion. Possible application scenarios of HRMs include online product ratings ("How do you assess the quality of the digital camera x?"), polls of expert judgments ("What is the probability of global warming to occur?"), psychological surveys ("Do you prefer red or white wine?"), or ratings that assess the quality of contributions in online communities ("Is the user comment of high quality?"). In these scenarios, explicit rewards can improve the quality of responses by encouraging a respondent to take the time to evaluate the question/contribution carefully, and to answer accurately and truthfully. Further, such rewards can potentially mitigate the under-contribution of ratings. However, appropriate rewards are difficult to determine because the objective truth is not available. This may be because the questions are inherently subjective (e.g., wine preference), or because the truthfulness of a response can only be established at a much later point in time (e.g., the occurrence of global warming). And simple rewards, for example a fixed remuneration per rating, are unlikely to yield the desired result. HRMs counter this problem by rewarding answers depending on the answers made by peers. They compute rewards in such a way that honesty, not conformity to the majority opinion, is the optimal strategy for respondents. (This does not exclude the majority opinion from being correct.) They achieve this by exploiting correlations between opinions of different persons regarding the same question.

2.2.1. The Peer Prediction Mechanism

In the following, we discuss the Peer Prediction mechanism (PP) [MRZ05]. First, we introduce the basic setting. Then we discuss different approaches for incentive compatible payments in this setting.

Setting

The peer-prediction method applies to settings where agents receive a noisy signal of some states of the world. The states of the world could be subjective preferences for

wine ('red/white') or the quality values of a digital camera ('low/medium/high' quality), etc. In the scenario discussed in this dissertation, the agents are the raters and the relevant states are the quality values of contributions made to a knowledge base, i.e., their correctness. In line with the literature, we refer to these states of the world as *types*.

The PP mechanism assumes that the type t of a given contribution is fixed, i.e., each contribution has a true quality that does not change over time. The number of types is finite and from the set of types T, i.e., $t \in T$. For example, a contribution could have types 'good' and 'bad'. The mechanism assumes that all raters share a common prior belief regarding the distribution of types $p(t)$. This common prior assumption is a standard assumption in game theory [SLB09]. The true type is hidden from the rater, who can only perceive a noisy signal. Let i denote a generic rater from the set of raters I, with $|I| \geq 3$. Each rater i privately receives signal S_i from the set $S = \{s_1, \ldots, s_m\}$ of possible signals. Think of the signal as the relevant information the rater uses to form her[1] subjective opinion about the quality of the contribution. (In case of the red wine question above, having a certain preference, e.g. 'I prefer red wine', constitutes the signal. For a setting regarding product ratings, experiencing the product quality constitutes the signal.) Let $P(s_j \mid t) = P(S_i = s_j \mid t)$ be the probability that a rater receives signal s_j when the true type of the product is t. In other words, each type induces a distribution of signals. The mechanism assumes that different types induce different signal distributions. E.g., a type 'low quality' might induce signals 'low quality' with higher probability than a type 'high quality'. Conditional on the true type of the contribution, signals are independent and identically distributed. The mechanisms assume $P(\cdot \mid \cdot)$ to be common knowledge, i.e., everyone knows that everyone knows, etc. [FT91].

The mechanism assumes that raters are rational, i.e., that they update their beliefs regarding priors in line with Bayesian reasoning. Upon receiving signal s_j, a rater updates her posterior belief that another randomly chosen rater receives signal s_k as follows

$$P(s_k \mid s_j) = \sum_{t \in T} P(s_k \mid t) \cdot P(t \mid s_j). \tag{2.1}$$

She obtains the posterior probability of type t given s_j (the second factor in the sum of Equation (2.1)) by applying Bayes' theorem

$$P(t \mid s_j) = \frac{P(s_j \mid t)p(t)}{P(s_j)}. \tag{2.2}$$

The prior signal belief (the denominator in Equation (2.2)) can be calculated from the conditional signal distribution and the type prior

$$P(s_j) = \sum_{t \in T} P(s_j \mid t)p(t). \tag{2.3}$$

Note that the inventors of the HRM claim, but do not test, that it is not necessary for users to do the rather complicated computations [MRZ05]. As long as they trust the mechanism to perform the updating correctly, users will prefer to report honestly.

[1] We refer to the rater who is scored by the HRM as female.

Example. For illustration, consider the following example. Assume, there are two possible types 'correct' and 'incorrect' for a contribution that we encode -1 and 1, respectively, i.e., $T = \{-1, 1\}$. Since it is assumed common knowledge, all raters (as well as the mechanism) know that the prior probability of a contribution to be of type 1 is $p(1) = 0.7$. Further, assume that there are two possible signals 'low' (l) and 'high' (h), with conditional signal distributions $P(h \mid -1) = 0.4$ and $P(h \mid 1) = 0.8$. Therefore, $P(h) = P(h \mid -1) \cdot P(-1) + P(h \mid 1) \cdot P(1) = 0.68$. Suppose that rater i has received signal h. In this case, according to Equation (2.1), her belief that another rater i' receives a 'high' signal is $P(S_{i'} = h \mid S_i = h) = 0.73$. Had rater i received a 'low' signal instead, her belief that i' receives a 'high' signal would be $P(S_{i'} = h \mid S_i = l) = 0.58$.

This subjective correlation between the signal of a rater and the signals of other raters is the essential piece of information that Peer Prediction (and other HRMs) exploits to incentivize truthful responses.

Empirical evidence for Bayesian updating. The assumption that respondents use their own opinion as evidence for the popularity of this opinion among others has been replicated in numerous studies, see [MM87] for an overview. For example, a red wine lover tends to estimate the ratio of people who prefer red over white wine higher than average. In general, respondents who endorse a certain opinion deem it more popular than those who do not. In one study, Ross et al. [RGH77] asked students to walk around the Stanford campus wearing a sandwich board[2] reading 'Repent!'. Students who agreed to engage in this activity estimated on average that 58.3 percent of all students would also agree. Students who were not willing to wear the board estimated on average only 29.7 percent would agree to wear it. We could also replicate the Bayesian-updating effect in our own study on how authors rate peer reviews they have received. Here, the more unfavorable ratings an author issues, the more he expects others to do the same. Initially, the literature has regarded this phenomenon as a 'false consensus effect', an egoistic bias to think that other people behave similar to us. Dawes, in particular, was the first to propose a Bayesian explanation for this phenomenon [Daw89, Daw90].

Incentive Compatible Scoring

After all raters have received the signals (i.e., formed their opinion), the mechanism asks them to submit ratings according to their signals. Let $r_i \in S$ denote the rating submitted by rater i. Which rating i actually submits upon receiving signal s_j is determined by her *rating strategy*. The rating strategy of rater i is a function $\sigma_i = (\sigma_i(1), \ldots, \sigma_i(m)) : S^m \to S^m$ that maps the signals received by i to the signals reported (i.e., her ratings). That is, i reports rating $\sigma_i(j) \in S$ whenever she receives signal s_j. The honest rating strategy is σ^* with $\sigma^*(j) = s_j$ for all $j \in \{1, \ldots, m\}$, i.e., the rater always reports the truth, that is, her rating equals the signal she received.

The mechanism scores each rater for submitting her rating. The mechanism computes the score by comparing i's rating to the rating of another rater, $ref(i)$, called the *reference*

[2] A sandwich board consists of two boards that hold a message and are connected by straps by which they are hung over a person's shoulders.

rater of i, with $ref(i) \neq i$. Let $\pi(\sigma_i(j), \sigma_{ref(i)}(k))$ be the score/payment received by i when she announces $\sigma_i(j)$ and $ref(i)$ announces $\sigma_{ref(i)}(k)$. The expected score ES of rater i depends on her posterior beliefs given the signal s_j she has received, her own rating strategy σ_i, and the rating strategy of her reference rater $\sigma_{ref(i)}$

$$\begin{aligned} ES(\sigma_i, \sigma_{ref(i)} \mid s_j) &= E_{s_k \in S}(\pi(\sigma_i(j), r_{ref(i)}(k)) \mid s_j) \\ &= \sum_{s_k \in S} P(S_{ref(i)} = s_k \mid S_i = s_j)(\pi(\sigma_i(j), r_{ref(i)}(k))), \end{aligned} \tag{2.4}$$

where the posterior signal belief $P(S_{ref(i)} = s_k \mid S_i = s_j)$ that $ref(i)$ receives the signal s_k is computed according to Equation (2.1).

The honest reporting strategy σ^* is a Bayesian Nash equilibrium (BNE) [Har67] if and only if for all signals $s_j \in S$ and all reporting strategies $\sigma' \neq \sigma^*$

$$ES(\sigma^*, \sigma^* \mid s_j) \geq ES(\sigma', \sigma^* \mid s_j). \tag{2.5}$$

That is, if $ref(i)$ reports truthfully, i's optimal strategy is to report truthfully as well. It is a strict BNE, if the above inequality is strict. Miller et al. prove that payment schemes $\pi(\cdot, \cdot)$ which satisfy Equation (2.5) for all raters exist [MRZ05]. The literature refers to such payment schemes, which make honest reporting a Bayesian Nash equilibrium, as Bayes-Nash *incentive compatible*. I.e., it is in every rater's best interest to truthfully report her signal if all the other raters report truthfully as well.

Payment Schemes of Classical Peer Prediction

The classical Peer Prediction (PP) mechanism [MRZ05] uses proper scoring rules to induce truthfulness. In the following, we briefly discuss scoring rules in general. Then, we show how the PP mechanism applies proper scoring rules to construct incentive compatible payments for ratings.

Proper scoring rules. Scoring rules [Win69, MW70, Sav71, Coo91, GR07] have been developed to elicit truthful probability predictions about events with publicly verifiable outcomes. In a nutshell, respondents are asked for the prediction of a future event, for example the probability of rain next Tuesday. After the event has materialized (on Wednesday) a scoring rule assigns a reward based on the predicted probability distribution and the materialized event.

More formally, a respondent announces a probability distribution $p = (p_1, \ldots, p_m)$ over m mutually exclusive events $\omega = \{\omega_1, \ldots, \omega_m\}$. Once the event ω_j has materialized, a scoring rule $R(p, \omega_j)$ scores the respondent based on his prediction p and ω_j. A scoring rule is proper if the respondent maximizes his expected score by announcing the prediction that corresponds to his true belief. It is strictly proper if the maximum is unique. The three best-known strictly proper scoring rules are

1. the quadratic scoring rule $R(p, \omega_j) = 2p_j - \sum_{k=1}^{m} p_k^2$,

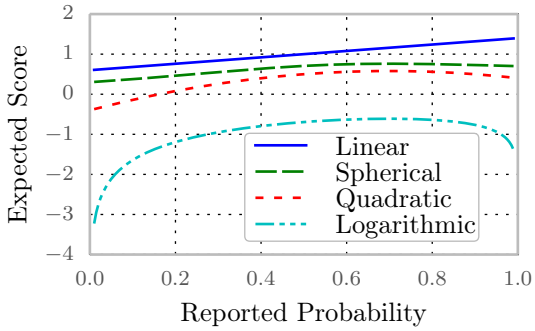


Figure 2.1.: Three strictly proper scoring rules and one improper scoring rule: expected scores for the true belief that Germany has a 70 percent chance of winning the FIFA world cup 2014.

2. the spherical scoring rule $R(p, \omega_j) = \dfrac{p_j}{\sqrt{\sum_{k=1}^{m} p_k^2}}$, and

3. the logarithmic scoring rule $R(p, \omega_j) = \ln p_j$.

Further, a strictly proper scoring rule remains strictly proper after any positive affine transformation. I.e., if $R(p \mid \omega)$ is a strictly proper scoring rule then $aR(p \mid \omega_j) + b$, $a > 0$, is also a strictly proper scoring rule.

As another example, suppose a respondent is asked to give a prediction for the following question: "What is the probability that Germany wins the FIFA World Cup 2014?". Suppose the respondent's true belief is that Germany has a 70 percent chance of winning. As Figure 2.1 visualizes, he can maximize his expected score by reporting his true belief. Contrast this with a scoring rule that is not proper: the linear scoring rule. The linear scoring rule rewards a respondent for the event that materialized with a score that is proportional to the probability assessment of that event: $R(p, \omega_j) = c \cdot p_j$, where c is some positive constant. Here, the respondent can maximize his expected score by setting his assessment to $p_j = 1$ if $p_j > p_k$ for $k \neq j$. That is, the linear scoring rule rewards overconfident statements.

Applying scoring rules to the Peer Prediction setting. In an HRM setting, the outcomes of events cannot be verified, either because they lie in the distant future or they are inherently subjective. Instead, the classical PP mechanism applies proper scoring rules to the posterior belief about the signal distribution of other raters. However, since signals are private information, the mechanism applies scoring rules to the ratings instead. Recall that $r_i \in S$ denotes the rating submitted by rater i. The PP mechanism scores every rater i dependent on the distribution over the reference ratings that her rating r_i predicts, i.e., $P(r_{ref(i)} \mid r_i)$, and on the actual rating submitted by her reference rater $r_{ref(i)}$

$$\pi(r_i, r_{ref(i)}) = R(P(r_{ref(i)} \mid r_i), r_{ref(i)}). \tag{2.6}$$

In other words, PP scores a rater on how accurately her rating predicts her reference raters' rating.

Miller et al. proof that honest rating is a strict BNE of the PP mechanism: if the rating of her reference rater is honest, a rater can uniquely maximize her expected score (Equation (2.4)) by truthfully announcing her subjective beliefs as well.[3]

Further, since strictly proper scoring rules remain strictly proper after positive affine transformation (see Section 2.2.1), the mechanism can scale payments without violating the Nash equilibrium conditions. Miller et al. show that it is possible to scale the payments in such a way that they cover the rating costs. Moreover, appropriate scaling can offset external benefits gained from lying, i.e., bribes.

Example continued. Recall, that after receiving a 'high' signal (h), rater i's posterior belief that the reference rater $ref(i)$ receives a 'high' signal is $P(S_{ref(i)} = h \mid S_i = h) = 0.73$. Conversely, had i instead received a 'low' signal (l), the posterior probability that $ref(i)$ receives h is $P(S_{ref(i)} = h \mid S_i = l) = 0.58$. We consider both cases in the following: (1) Assume, she has received h. Using the logarithmic scoring rule, the resulting expected score for i if she reports h is

$$ES(h, h \mid h) = 0.73 \log 0.73 + 0.27 \log 0.27 = -0.58.$$

If she lies and reports l instead, the mechanism scores her according to the posterior distribution implied by l. In that case, her expected score is

$$ES(l, h \mid h) = 0.73 \log 0.58 + 0.27 \log 0.42 = -0.64.$$

Now, (2) if she had received l instead, her expected payoffs were -0.68 and -0.74 for announcing l and h, respectively. So telling the truth, i.e., announcing the signal she has received, maximizes her expected payoff in both cases.

As noted above, the inventors of the HRM claim that it is not necessary for users to do the rather complicated computations [MRZ05]. As long as they trust the mechanism, users will prefer to report honestly.

2.2.2. Peer Prediction with Linear Programming

Jurca et al. [JF06] use automated mechanism design [CS02] to construct incentive compatible payments. That is, instead of applying proper scoring rules, they define payments as an optimization problem. In order to achieve this, they redefine the BNE (Equation (2.5)) conditions, adding two aspects. First, they explicitly consider the rating costs, i.e., the costs for evaluating a contribution and for submitting the rating. Second, they consider the potential benefits a rater might gain from lying. Then they use these conditions as constraints of a linear program that minimizes the expected amount of money the mechanism has to pay.

[3] To guarantee the strictness of the BNE, the proof assumes stochastic relevance [MPZJ07] for signals, i.e., raters with different signals have different posterior beliefs. Stochastic relevance is almost always satisfied when different types generate different signal distributions [MRZ05].

Let $c \geq 0$ denote the fixed rating costs. Further, let $\lambda(s_j, \sigma'(j))$ be the external benefit a rater can gain from reporting signal $\sigma'(j)$ when she observed signal s_j, where $\lambda(s_j, s_j) = 0$ and $\lambda(s_j, s_l) \geq 0$ for all $s_j, s_l \in S$ and $s_j \neq s_l$. Incorporating the rating costs and the lying benefits into the BNE of Equation (2.5) yields

$$\begin{aligned} &ES(\sigma^*, \sigma^* \mid s_j) \geq ES(\sigma', \sigma^* \mid s_j) + \lambda(s_j, \sigma'(j)), \\ &ES(\sigma^*, \sigma^* \mid s_j) \geq c. \end{aligned} \tag{2.7}$$

This means that the honest rating strategy must yield a higher expected payoff than lying and that this expected payoff must exceed the rating costs c. Given c and $\lambda(\cdot \mid \cdot)$, the honest reporting strategy is a BNE if and only if the inequalities in Equation (2.7) hold for all signals $s_j \in S$ and all reporting strategies $\sigma' \neq \sigma^*$.

Using the definition of the expected score (Equation (2.4)) the BNE conditions above can be expressed as follows

$$\begin{aligned} \sum_{s_k \in S} P(s_k \mid s_j)(\pi(s_j, s_k) - \pi(s_l, s_k)) > \lambda(s_j, s_l), \\ \sum_{s_k \in S} P(s_k \mid s_j)\pi(s_j, s_k) > c, \end{aligned} \tag{2.8}$$

for all $s_j, s_l \in S$, $s_l \neq s_j$.

The expected amount the mechanism has to pay an honest rater is the sum of the expected payments for that rater weighted by the prior probabilities of the signals he can receive

$$\mathrm{E}_{s_j \in S}\left(ES(\sigma^*, \sigma^* \mid s_j)\right) = \sum_{s_j \in S} P(s_j) \left(\sum_{s_k \in S} P(s_k \mid s_j)\pi(s_j, s_k) \right). \tag{2.9}$$

The optimal payment scheme $\pi(\cdot \mid \cdot)$ minimizes the expected cost of the mechanism (Equation (2.9)) while satisfying the conditions of the BNE (Equation (2.8)). Thus, the optimal payment scheme solves the following linear program

$$\begin{aligned} \text{minimize} \quad & \sum_{s_j \in S} P(s_j) \left(\sum_{s_k \in S} P(s_k \mid s_j)\pi(s_j, s_k) \right) \\ \text{subject to} \quad & \sum_{s_k \in S} P(s_k \mid s_j)(\pi(s_j, s_k) - \pi(s_h, s_k)) > \lambda(s_j, s_l) && , \forall s_j, s_l \in S, s_j \neq s_l \\ & \sum_{s_k \in S} P(s_k \mid s_j)\pi(s_j, s_k) > c && , \forall s_j \in S \\ & \pi(s_j, s_k) \geq 0 && , \forall s_j, s_k \in S. \end{aligned} \tag{2.10}$$

Example with LP. We continue the the two signals, two types example from above and set the lying benefits to $\lambda(l, h) = \lambda(h, l) = 0.05$ and the rating costs to $c = 0.1$. The optimal incentive compatible payments $\pi(r_i, r_{ref(i)})$ in that case are $\pi(h, h) = 0.2252$, $\pi(l, l) = 0.4224$, and $\pi(h, l) = \pi(l, h) = 0$.

2.2.3. Lying and Collusion Equilibria

Even though honest reporting is the desired equilibrium strategy of the Peer Prediction mechanism, it is not unique. Other equilibria like rating always high or rating always low exist as well. [JF07] proposes countermeasures against such lying coalitions. The countermeasures are based on increasing the number of reference ratings per rating and increasing the budget of the mechanism to offset incentives for such lying coalitions. Systematic taste differences among raters also pose a potential threat to the proper functioning of the mechanism. For example, some raters might have contrarian views or might generally be harsher in their assessment of quality. Whether the problems of lying equilibria or taste differences occur without countermeasures in reality is an interesting question that our experiments will address as well.

2.2.4. Further Honest Rating Mechanisms

In this section we briefly discuss HRMs similar to the Peer Prediction mechanism that relax some of its assumptions.

The Bayesian Truth Serum

The Bayesian Truth Serum (BTS) [Pre04] also makes truth-telling a BNE. BTS makes basically the same assumptions as Peer Prediction (Section 2.2.1). In particular, it assumes rational (i.e., Bayesian) updating from a common prior shared between all raters. However, in contrast to PP the common prior needs not be known by the mechanism. In addition to her rating, the BTS mechanism asks each rater for her posterior signal beliefs. That is, for each rated contribution each rater also predicts the empirical distribution of all ratings for the contribution. The mechanism scores the predictions for accuracy. Additionally, it assigns an 'information score' to each rating $r_i = s_j$ that reports signal s_j as follows

$$\text{information score for } r_i = \log \frac{\text{actual relative frequency of } s_j}{\text{(geometric) mean predicted frequency of } s_j}.$$

Intuitively, ratings that are more common than collectively predicted receive high scores while ratings that are less common receive low scores. The resulting BNE only holds for sufficiently large numbers of raters.

For our scenario of ratings in an online community we deem BTS not suitable because of its prohibitively high rating costs. As we have seen above, ratings are already sparse in most online communities. Our experiments show that they remain sparse (though considerably less so) even after applying an HRM. Expecting every rater to issue one rating plus one prediction of the empirical distribution of ratings for *every* rated contribution is rather unrealistic.

Recent Extensions of Peer Prediction and the Bayesian Truth Serum

Since we have conducted the experiments in this dissertation, a number of extensions of PP and BTS have been proposed. [WP12b] introduces a 'robust Bayesian Truth Serum'

(RBTS), an extension of the BTS mechanism that does not require large numbers of raters. Instead, it is incentive compatible for populations consisting of more than two raters and works for binary signals only. Like the original BTS, the RBTS asks the rater for two reports, a prediction and a rating, which it scores by means of strictly proper scoring rules. [RF13] also introduces an extension of the BTS that works for small populations with more than 1 rater and for categorical signals. It elicits two reports for each rated contribution as well. Finally, [WP12a] extends Peer Prediction such that it works without a common prior. To achieve this, it elicits two ratings: one rating before the rater has seen the contribution and one afterwards.

Like BTS, these new mechanisms also have high rating costs compared to the original PP mechanism. This makes them likely unsuitable for the scenario envisioned as well. However, they could be evaluated in future work in settings where high rating costs are more tolerable, e.g., in settings with only few ratings overall. An example of such a setting could be an expert team that creates a 'seed' ontology with very few items whose correctness has an over-proportional influence.

2.3. Structured Knowledge and Ontologies

In Section 1.1, based on [NCA08], we introduced the term *structured knowledge* as referring to any kind of interrelated, conceptualized information. This can range from a set of terms with informal relationships, like a hierarchy of tags, to a fully axiomatized ontology. In this section, we discuss the basic notion of ontology and related terms, and clarify their usage.

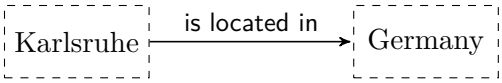


Figure 2.2.: Structured knowledge: two-node network.

To gain intuition about structured knowledge and its usefulness, let us start with a simple example. Consider the sentence 'Karlsruhe is located in Germany'. We can break this apart into the two terms `Karslruhe` and `Germany` and the relationship `is located in`, as depicted in Figure 2.2. This form makes the two entities (Karlsruhe and Germany) and their relationship explicit and thus easier to interpret for a computer than the natural language version. It is also an abstraction that captures the essence of many similar formulations of the same fact ('Karlsruhe is a city in Germany', etc.).

Adding more semantics to this structure will make it more useful for (human or machine) interpretation and automatic processing. For example, we can tell the computer that Karlsruhe is a city and that Germany is a country. Further, we can enhance the information structure by generalizing the concepts `City` to `Settlement` (which also includes `Village`), and adding attributes such as `founding date`. Figure 2.3 depicts the resulting ontology. Based on this ontology, an automatic agent that is looking for a vacation destination in Germany on our behalf could identify Karlsruhe as a potential destination.

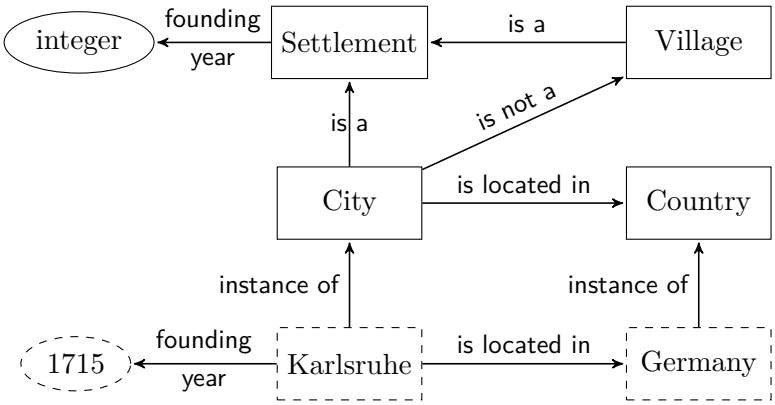


Figure 2.3.: Simple city-country ontology.

Note that the semantics in Figure 2.3 are represented in the structure. In the words of Tim Berners-Lee [BLF99, p. 12]:

> [...] a piece of information is really defined only by what it's related to, and how it's related. There really is little else to meaning. The structure is everything.

In the following, we give a brief overview of ontologies and related forms of structured knowledge. For a more detailed discussion see, for example, Staab and Studer [SS09] or Domingue et al. [DFH11].

2.3.1. Definition of Ontology

The word 'ontology' comes from philosophy where it refers to the science of being. Ontology, as a discipline of philosophy, studies the kinds and structures of objects, properties, and events in every area of reality. One of ontology's earliest practitioners was Aristotle[4] as documented in his book *Metaphysics*. The term 'ontology' itself was coined much later, namely in the seventeenth century, independently by two German philosophers. See [Flo04] or [BS91] for more detailed discussion of ontology in philosophy. Computer science takes a more pragmatic stance. Here, *an* ontology refers to an artifact that encodes knowledge about a domain of interest. In computer science, ontologies are in particular used as knowledge representation formats in artificial intelligence [RN13] and play a key role in enabling the Semantic Web [BLHL01].

A common definition of ontology in computer science is the following [GAVS11] (cf. [SBF98, Gru93] for similar definitions):

> An ontology is a formal explicit specification of a shared conceptualization of a domain of interest.

The definition entails the following:

[4] Aristotle himself called it 'first philosophy' [Flo04, p. 155].

- A *Conceptualization* is an abstract, simplified view of the world. The view is represented by objects or concepts and the relationships between them [GN87, GG95].
- *Formal* means that the ontology must be serializable in some machine readable knowledge representation language that exhibits well-defined semantics.
- *Explicit* refers to the fact that concepts, relationships, and constraints are explicitly defined and thus accessible for machines.
- *Shared* means that the members of the community using the ontology all agree to a sufficient degree upon the conceptual model contained within the ontology. Without such a consensus, the ontology might be practically useless [GOS09].
- Finally, the conceptualization is typically limited to a specific *domain* of interest. This simplifies its creation and maintenance, as well as its reusability.

2.3.2. Components of an Ontology

The following components are characteristic for ontologies [GAVS11, UG04]:

- *Concepts.* Ontologies contain a collection of terms that are of interest in a given domain. General concepts of the domain are captured in form of classes (also called types or categories) such as `City` or `Country`. Further, ontologies might contain individuals, i.e., instances of classes, for example `Karlsruhe` as an instance of `City`.
- *Relationships.* By connecting concepts, relationships provide meaning and structure to the ontology. Without them, the ontology would simply be a collection of terms. In Figure 2.3, `is located in` is a relationship.
- *Class-instance relationships.* They are the essential means to distinguish between classes and instances. At the same time, they associate individual instances with classes. For example, `Karlsruhe` is an instance of the class `City`.
- *Superclass-subclass relationships,* also called subsumptions or *is-a* relationships. Classes can subsume other classes. The subsuming class is called superclass (or supertype) while the subsumed class is called subclass (or subtype). For example, `City` is a subclass of `Settlement`. In general, the subclass implicitly inherits the properties of the superclass. For example, `City` inherits `founding year` from `Settlement`.
- *Attributes.* These are properties of classes or instances with simple data types, such as string, or integer. For example, a `Settlement` has a `founding year` of type integer.
- *Axioms.* Axioms are logical statements that say what is true in a given domain. In principle, axioms can specify arbitrarily complex rules. The most commonly used axioms are the relationships listed above. Further typical axioms include

constraints for the domain and range of properties, the transitivity, reflexivity and symmetry of relations, and the disjointness of classes. In Figure 2.3, we could define the `is located in` relation as transitive, and asymmetric, or specify that `Village` and `City` are disjoint classes.

Formal axioms allow for deductive reasoning. Deductive reasoning can for example infer new facts that are only implicitly stated in the ontology. Further, it can verify an ontology, i.e., ensure that it does not contain contradictory information.

An ontology including a set of instances constitutes a *knowledge base* [UG04].

Components such as classes, relationships, constraints, or subsumption are also part of other conceptual modeling techniques used in computer science such as entity-relationship models (ERMs) [Che76] or the Unified Modeling Language (UML) [RJB04]. In fact, as we will see below, models based on ERMs and UML class diagrams can be regarded as a form of ontology. The difference between the models commonly created with these techniques and ontologies in the narrow sense is subtle and lies in the usage and the purpose, as well as in the expressivity [GAVS11, UG04]. The main purpose of UML class diagrams and ERMs is designing software systems or database schemas, respectively. As such, the models are focused on representing only the concepts relevant to the respective software system, and thus they often incorporate technical design decisions. Ontologie's (in the narrow sense) main purpose, on the other hand, is to serve as a source of domain knowledge. Hence, the resulting models are more general. Further, ontology languages allow for more expressive constructs that are typically not found in ERMs and UML, and are especially designed to allow for deductive reasoning.

2.3.3. Usage of Ontologies.

Grüninger and Lee [GL02] identify three different usage areas of ontologies: communication between agents (human or automatic), computational inference, and reusing and organizing knowledge (e.g., structuring digital libraries).

In the following, we discuss four different areas of usage of ontologies based on Grimm et al. [GAVS11]:

- *Knowledge organization.* Many information systems contain representations of knowledge that are difficult to process automatically, such as unstructured or only partially structured data like books, newspaper articles, images, or other multimedia documents. Here, ontologies allow for structuring and organizing the stored knowledge [GL02]. Further, ontologies provide semantics to metadata for web resources.

- *Semantic search.* This includes queries to a single knowledge base ("Find the five longest rivers running through German cities founded before 1700."), as well as semantically enhanced search over the World Wide Web. In the latter case, for example, ontologies provide background knowledge for concept-based query expansion [BMS07], or for ranking query results according to the "rational surfer model" [FDP+05].

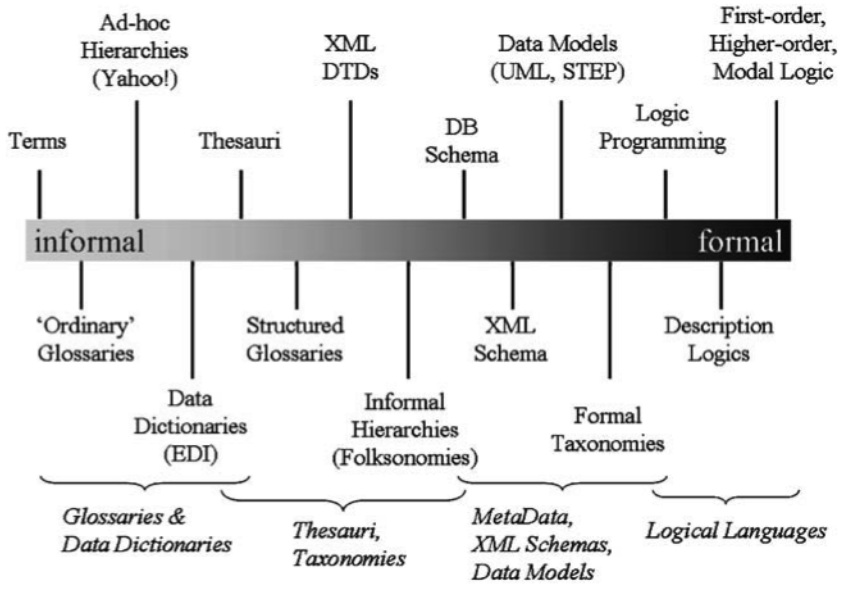


Figure 2.4.: Example of an ontology spectrum by degree of formality by [GOS09].

- *Integration.* According to Uschold and Grüninger, ontologies can act as a neutral template for data models used in multiple applications. Moreover, ontologies can serve as an interchange format mapping between the data models of different applications [UG04].
- *Formal processing and inference.* Automated reasoning over ontology-based background knowledge is a functionality that may be applied in the usage areas above.

In the future, ontologies may support agent-based scenarios such as sending a software agent to buy a laptop computer, including searching multiple sites and comparing offers over these sites [RN13], or letting the agent arrange schedules for physical therapy sessions [BLHL01].

2.3.4. Ontology Spectra

We can divide ontologies by their degree of formality into a spectrum starting with *lightweight ontologies* at one end and *heavyweight ontologies* at the other [UG04, GAVS11]. Here, degree of formality refers to the extent to which the ontology is axiomatized. Lightweight ontologies possess no or only a few axioms, and may consist of terms only, with little specification of the meaning. Heavyweight ontologies are characterized by extensive axiomatization. Along the spectrum, parallel to the degree of formality, the amount of meaning specified, and the support for automated reasoning increases [UG04]. Figure 2.4 depicts one such spectrum. Similar spectra exist that classify the various ontology subtypes according to their degree of formality [UG04, McG05], as well

as according to similar dimensions. These dimensions include support for automatic reasoning [SW01], semantic interoperability [Obr03], and expressiveness [Wel07]. Besides their dimension, the spectra differ regarding their level of granularity. Moreover, they differ regarding the relative order of the different ontology subtypes. For example, some authors place thesauri along the formality continuum to the left of taxonomies while others place them to the right. Further, some authors place folksonomies on the far left side of the formality spectrum [Wel07] while others place them right of thesauri [GAVS11, GOS09]. See [Bra11] for a comprehensive discussion of ontology spectra.

In the following, we describe important types of ontologies along the formality spectrum loosely based on [GAVS11]:

- *Glossaries and folksonomies* are unstructured, uncontrolled lists of keywords or tags. Glossaries list terms and their meanings in an unstructured way. Folksonomies are a collection of tags (keywords) assigned to resources such as websites (e.g., at Delicious), images (Flickr), or videos (Youtube). As opposed to conventional keyword indexes used for example to annotate books in library catalogs, folksonomies arise from collaborative tagging activities of many users and thus add a social dimension.

- *Thesauri* are a simple form of structured knowledge that organize the words of a domain according to lexical criteria such as synonymy, antonymy, homonymy, etc. Thus, their formality and expressivity are rather low. Examples are the Historical Thesaurus of the Oxford English Dictionary [hto] or the WordNet thesaurus [Fel98].

- *Concept schemes* are semantic structures that are characterized by rather informal semantic relationships, such as ad-hoc hierarchies like the Yahoo catalog.[5] Typically, concept schemes arise from collaborative efforts of larger communities. They also include tag hierarchies generated from folksonomies by exploiting the co-occurence of tags, or the similarity of tag usage by different users [Mik05, HGM06].

- *Formal taxonomies* are class hierarchies based on a formal notion of subsumption (i.e., the is-a relationship is transitive).

- *Conceptual Data Models* such as UML class diagrams or ERMs are aimed towards designing information systems (see discussion above). In these models, logical formalization is typically used to check constraints rather than for deductive reasoning.

- *Rule and fact bases* are knowledge bases consisting of a large number of instances. They typically exhibit basic querying and simple reasoning capabilities over class and property hierarchies.

[5] [LM01] give an example of an informal is-a relationship taken from the yahoo catalog: Here, the general category 'apparel' includes a subcategory 'women's apparel' that includes subcategories 'dresses' and 'accessories'. While every instance of 'dress' is an instance of 'apparel', 'fragrance' (which is an instance of 'accessories') is not an instance of 'apparel'.

- *General logical theories* are the most formal type of ontology. They exhibit a highly axiomatized and expressive knowledge representation. They formulate axioms in first order, higher order, or modal logic.

For ontologies based on logical languages, there exists a trade-off between expressiveness and reasoning efficiency [GOS09]. The very expressive formal languages do not allow for sound and complete reasoning, are in general undecidable. Less expressive subsets of first-order logic, like description logic, on the other hand, allow for decidable and efficient reasoning.

Moreover, and more important from the users' perspective, there exists a trade-off between the degree of formality/expressiveness of an ontology and the size of its community. The more formal and expressive an ontology the higher the cost for understanding it and thus the smaller its community [Hep07]. Further, lightweight ontologies are easier for users to understand and thus better suited for to be created and maintained by communities than heavyweight ones.

2.3.5. Ontology Languages

There exists a variety of ontology representation languages, in particular in the context of the Semantic Web. We only give a brief overview of some prominent examples and refer to [AVH04, PRH+06] for a detailed discussion. Note that in this thesis, we are primarily concerned with the design and deployment of incentive mechanisms for the collaborative construction of structured knowledge. In this specific context, the ontology language to encode the knowledge is of secondary concern.

RDF(S). The Resource Description Framework (RDF) is a standard developed by the World Wide Web Consortium (W3C) to encode metadata and ontologies for the Semantic Web. To make statements about (web) resources, RDF uses subject-predicate-object triples. To allow for a worldwide unique identification, subjects, predicates, and objects are usually identified by *uniform resource identifiers* (URIs). Thus, the statement 'Karlsruhe is located in Germany' can for example be expressed as (subject, predicate, object) triple:

```
(http://dbpedia.org/resource/Karlsruhe,
http://www.geonames.org/ontology#locatedIn,
http://dbpedia.org/resource/Germany)
```

RDF Schema (RDFS) is defined on top of RDF. As basic ontology language, RDFS includes features such as relationships, instantiation, subsumption.

OWL. The Web Ontology Language (OWL) is a W3C standard defined on top of RDF(S). It serves as the main ontology language for the Semantic Web. OWL offers a high expressiveness. Besides the capabilities inherited from RDF(S), OWL is able to express for example disjointness between classes, boolean combination of classes (for example, a union of classes `Male` and `Female` to `Person`), and special restrictions for

relationships such as the relationship's cardinality, transitivity, or reflexivity. To account for the trade-off between expressiveness and efficient reasoning (see page 30), OWL comes in several variants with varying degrees of expressiveness: For example, OWL Full is fully compatible with RDF(S), both syntactically and semantically. OWL DL, is a more restricted version of OWL, designed for efficient reasoning.

Topic Maps. Topic Maps [Pep00] are an ISO standard, and define a data model and an associated data format for knowledge representation. Topic Maps can be categorized as lightweight ontologies residing roughly on the level of concept schemes. In contrast to RDF(S) or OWL, which are built for automatic processing, Topic Maps provide few and easy to understand semantic features. This makes them well-suited for direct manipulation by humans. The basic elements of Topic Maps are topics, associations, and occurrences. Topics represent real or abstract entities (like `Karlsruhe` or `Germany`) and can be grouped together by means of topic types. Topics and topic types roughly represent instances and classes, respectively. Associations represent relationships between topics and can be grouped together by association types. Topics that partake in associations have assigned association roles (for example, `Karlsruhe` could play the `contained` role while Germany could play the role `container` in a `located in` association). Occurrences link topics to external resources. Like RDF, Topic Maps use URIs to uniquely identify elements.

SKOS. The Simple Knowledge Organization System (SKOS) is another language for representing lightweight ontologies. It was announced 2009 by the W3C and is build on top of RDF(S).[6] SKOS's fundamental modeling element is `Concept`. Further, among other modeling primitives, SKOS allows for defining labels for concepts, and provides two kinds of relationships: broader/narrower relationships and associative relationships.

2.3.6. Traditional Ontology Engineering

Ontology engineering refers to the "activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies" [GPCFL04]. A number of methodologies for engineering ontologies from scratch have been proposed. They can broadly be categorized into traditional and collaborative ones. The traditional methodologies mainly follow a centralized approach where a small number of ontology engineering experts develop the ontology, using the input elicited from domain experts. See [GPCFL04] for an overview. The traditional methodologies are somewhat influenced by traditional software engineering methods. Pinto and Martins [PM04] identify five different stages of the (traditional) ontology engineering process (see also [GPCFL04, ST06]): (i) specification, i.e., identifying the purpose and scope of the ontology, (ii) conceptualization, i.e., creating the conceptual model, (iii) formalization, i.e., transforming the conceptual model into a formalized form,

[6] `http://www.w3.org/TR/2009/REC-skos-reference-20090818`

(iv) implementation of the model into a knowledge representation language, and (v) maintenance, i.e., updating and correcting the ontology.

The problem with these traditional methodologies is that they insufficiently consider the question that is essential for the usefulness of an ontology, namely, how to reach a shared consensus among all stakeholders of the ontology. Instead, they put the knowledge engineers in a central position, while only sparely, or not at all, involving the actual users of the ontology in the creation process [VPTS05]. This centralized approach also makes ontology creation and maintenance costly and time consuming activities, because they are executed by rare and expensive ontology engineers. Thus, nowadays "it is generally acknowledged that, in order to be useful, but also economically feasible, ontologies should be developed and maintained in a community-driven manner" [SLR14].

We discuss collaborative ontology engineering methodologies in Section 3.1.2.

2.3.7. Ontology Learning

The manual engineering of ontologies is a laborious endeavor. Therefore, various techniques for the automatic or semi-automatic generation of ontologies from various kinds of data sources have been developed. These techniques are summarized under the label 'ontology learning'. See [CMSV09, WLB12] for a detailed discussion. [WLB12] (based on [BM05]) groups the aspects of ontology learning into sub-tasks of increasing complexity. These sub-tasks are the learning of: terms, concepts, taxonomic relations, non-taxonomic relations, and axioms. This group of sub-tasks is also called the ontology learning layer cake. To accomplish these sub-tasks, ontology learning leverages techniques from machine learning or computational linguistics. For example, for extracting terms from natural language text, linguistic techniques such as part-of-speech tagging [JM09] can be employed. Statistic techniques such as tf-idf [MRS08] can filter the relevant terms in a subsequent step. To assign terms into groups, discover concepts, or construct hierarchies clustering techniques (among others) can be employed [WLB12]. Besides using text sources, there are also approaches to automatically extract ontologies from semi-structured sources such as UML class diagrams [XNH+11], tables [TEL+05], or folksonomies [DHS07]. Information extraction approaches such as [SSW09, ECD+04] extend existing (usually handcrafted) ontologies by parsing natural language documents, and extracting ontological facts from them. Some of these focus in particular on extracting structured knowledge from Wikipedia [LIJ+14, SKW07].

Despite these advances, the formal quality of ontologies generated with ontology learning is still insufficient for many practical applications [GAVS11]. Even though automated or semi-automated approaches to ontology learning can support the ontology engineering process to a large degree, ontology development remains a human-driven process [SS10].

3. Peer Production of Structured Knowledge – Empirical Studies of Rating-Based Incentive Mechanisms

The question of how to create structured knowledge continues to be a fundamental research issue. Experts for its creation are rare and costly, and automatic solutions remain insufficient for many practical applications (cf. 'Traditional Ontology Engineering' on page 31 and 'Ontology Learning' on page 32).

In this chapter, we study how to utilize web-based peer production for the creation of structured knowledge. This is a promising approach. It shifts the burden for creating the structured knowledge away from the rare experts and puts it into the many hands of peers within online communities. Compared to traditional centralized methods, this has several advantages: it inherently fosters consensual agreement about the semantics captured in the knowledge structures, is less costly, and reduces the time lag between changes in knowledge and adapting the knowledge representation.

However, peer production of structured knowledge poses the same challenges as peer production in general (see Chapter 1) namely: (1) How to motivate peers? As discussed in Section 2.1.1, many online communities suffer from under-contribution, i.e., only a minority contributes. (2) How to ensure and assess data quality? Quality assurance is in particular important for structured knowledge, which is used for query processing or automated reasoning.

To answer these questions, we study how rating-based incentive mechanisms can motivate users and assure the quality of the structured knowledge created collaboratively. We assume the following real-world scenario: An online community creates structured knowledge. Its members review the contributions of each other and rate them according to the quality perceived. To motivate users to contribute, they receive points according to the number and quality of their contributions. The quality of contributions, in turn, is computed based on the ratings they receive. Finally, the points gathered are converted into external rewards, e.g., gift coupons as with Epinions or system privileges as with Slashdot.

Since ratings play such a crucial role – they measure contribution quality and determine rewards for good contributions – they have to be of high quality as well. To motivate users to provide high-quality ratings, we apply an HRM to our scenario, namely the Peer Prediction mechanism, described in Section 2.2.1. We have chosen Peer Prediction because it has the lowest rating costs among all HRMs. For Peer Prediction, a user needs to submit only one rating for a contribution she wants to assess. Other HRMs require two ratings instead, or even estimates of the distribution of other ratings. (For a

discussion of the rating costs of HRMs see Section 2.2.4). This makes Peer Prediction best-suited for our scenario. However, despite the various articles on HRMs, not much effort has been expended to test these mechanisms empirically. In fact, we are only aware of one empirical study of an HRM, namely of the Bayesian truth serum [WP13]. In the study, the authors test the mechanism by means of questionnaires. In particular, we are not aware of any studies that apply HRMs in the context of collaborative knowledge creation tasks within an online community.

We study the following research questions regarding the creation of structured knowledge:

1. How do reward mechanisms for contributions as well as for ratings influence user behavior?

2. How do rewards for contributions that are either fully or partially dependent on ratings influence the quality and the number of contributions, compared to rewards that are fixed?

3. Does an HRM lead to ratings of higher quality, compared to a fixed reward per rating?

To this end, we have developed a platform for the collaborative creation of structured knowledge called *Consensus Builder* (CB). It provides the functionality to browse the knowledge base and to create and change data items. To address the problems of low quality and under-contribution of ratings, CB allows for a fine-grained rating of the semantic structures. Further, it features rating-based incentive mechanisms, in particular an HRM. The underlying incentive mechanisms can easily be exchanged. This enables us to investigate to which extent these mechanisms stimulate the quality and the quantity of the data. CB is operational in a real-world environment.

Based on the research questions, we formulate a number of hypotheses. We extensively test the hypotheses in a series of controlled field experiments using CB as a platform. The experiments took place in different settings and with participants of different backgrounds, using setups close to the real-world scenario envisioned.

A main finding of ours is that an HRM leads to ratings of equal or higher quality, compared to static rewards. Further, we find that fully rating-dependent – but not partially rating-dependent – rewards for contributions improve contribution quality, but result in fewer contributions compared to a fixed reward per contribution.

The chapter is structured as follows. Section 3.1 reviews related work. Section 3.2 presents our collaboration platform Consensus Builder. Section 3.3 discusses our design decision regarding the application of the HRM. Section 3.4 states our hypotheses. Section 3.5 introduces the experimental setups, and Section 3.6 present the results of the experiments. Section 3.7 discusses the results and the lessons learned in the experiments, and gives some design recommendations for structured-knowledge-creating communities. Section 3.8 concludes.

3.1. Related Work

3.1.1. Extrinsic Incentives for Contribution Quality

In Section 2.1 we have reviewed the literature on motivation in peer production in general and in online communities in particular. As we have stated in Section 2.1.5, we are not aware of any work that experimentally tests extrinsic incentives to rate the quality of contributions.

3.1.2. Collaborative Methods for the Creation of Structured Knowledge

Section 2.3.6 introduced traditional, centralized, ontology engineering and concluded that generally ontologies should be developed in a community-driven manner. Further, Section 2.3.7 discussed various techniques for creating ontologies automatically. It concluded, that the formal quality of ontologies generated with ontology learning is still insufficient for many practical applications and that ontology development remains a human-driven process.

In the following, we discuss methodologies that enhance the traditional ontology engineering approach with collaborative aspects. Beyond that, we discuss how to construct structured knowledge with alternative approaches such as using paid crowdsourcing like Amazon Mechanical Turk, or using gamification[1] to motivate users.

Collaborative Ontology Engineering Methodologies

Several methodologies for ontology engineering that incorporate collaboration mechanisms have been proposed [HJ02, MLMS06, VPTS05, KV06, AH07]. Most of these methodologies aim to develop heavyweight, fully axiomatized ontologies. They differ, for example, w.r.t. the exact strategy and the techniques used for creating the ontology, or w.r.t. the kind of users involved in the creation process (knowledge workers, domain experts, ontology engineers, ontology users). [SLR14] provides a comprehensive overview.

Holsapple and Joshi were the first to propose a collaborative methodology for ontology engineering [HJ02]. Their method consists of four phases: In the preparation phase, knowledge workers specify design criteria and related standards to guide the development of the ontology. In the so called anchoring phase, the knowledge workers develop an initial ontology to seed the collaborative effort. In a third phase, a panel of knowledge workers iteratively improves the ontology using a process model adapted from the Delphi method [LT75]. Views on the ontology are elicited from each participant individually, and in a later step reconciled by means of feedback, thus fostering a consensual understanding. In the fourth phase, the ontology is deployed for actual usage.

The DILIGENT methodology [VPTS05] involves domain experts, ontology engineers, knowledge engineers (roughly: mediators between ontology engineers and domain experts), as well as ontology users. A small group consisting of domain experts and knowledge/ontology engineers creates the core ontology. This ontology is subsequently

[1] Gamification is the use of game design elements in non-game contexts [DKND11].

used and updated locally by ontology users. A central board (presumably) consisting of knowledge and ontology engineers is responsible for controlling and updating the ontology. The engineering board collects the local changes made by users, analyses them, and revises the ontology accordingly.

The HCOME methodology proposes a more human-centered, decentralized engineering model [KV06]. Here, knowledge workers discuss requirements for a shared ontology during a specification phase. Afterwards, they develop ontologies individually, and, in a later step, share them with the community for revision and merging.

Most of the collaborative methodologies for ontology engineering develop heavyweight ontologies not suitable for direct manipulation by end users alone. Further, these methodologies, with the exception of HCOME, rely on centralized techniques using ontology engineers to some degree. Most importantly, neither of the methodologies above is explicitly concerned with mechanisms for user motivation.

Braun and colleagues introduce a more light-weight, Web 2.0 based methodology for ontology engineering [BSW+07]. Their community-driven approach relies on the maturing of ideas and concepts from tags into formal ontologies. Further, they discuss intrinsic motivations users might have to engage in the maturing activities, such as future retrieval or contribution and sharing. The maturing starts with the reuse of tags within the community, which leads to the emergence of shared vocabularies and the convergence of tags to concepts. In a later step, for example, users create class hierarchies because they want to better support their search needs or because they like to describe concepts or images in more detail, etc. Their approach might benefit from explicitly incentivizing users, e.g., using the mechanisms presented and evaluated here.

Various tools for the collaborative creation of structured knowledge have been proposed, ranging from full-fledged ontology editors with collaborative features [SKKM03, TN07] over Wiki-based approaches for semantic data [AD06, CCT04, Sou05, Sch06, VKV+06] to tools that support tagging folksonomies [J+07, ZB07]. Some of these tools feature rating mechanisms. All this work does not include systematic attempts to evaluate the effect of ratings on knowledge quality for these tools. There also exist commercial tools for the collaborative creation of structured knowledge, notably Freebase [BEP+08]. Noy et al. [NCA08] give a detailed overview of tools for the collaborative construction of structured knowledge. Further, they discuss general requirements for such tools and the different collaboration approaches implemented in these tools.

Siorpaes and Simperl analyze tools and methodologies for semantic content creation, including the ones described above, and identify tasks that are inherently human driven [SS10], i.e., tasks that require human input and are not automatable. For example, they identify describing the domain and scope of ontologies, defining axioms, building class hierarchies, and creating class-instance relationships as human driven tasks. Collecting relevant terms and discovering suitable ontologies for reuse are examples of tasks that they identify as partially automatable.

Paid Crowdsourcing and Gamification

Eckert et al. use input from Amazon Mechanical Turk (AMT) workers to construct *is-a* relations between pre-selected terms in a philosophy knowledge domain [E+10]. They propose a redundancy-based method to achieve high-quality results from the input of AMT workers. As opposed to our approach, they included concept pairs for which they could objectively determine a correct answer, i.e., a gold standard, to identify well-performing AMT workers. Further, their method, as any AMT-based scheme, is only applicable to domains known to the general public. Only such domains can attract a sufficient number of AMT workers with the necessary understanding. For example, AMT workers do not possess the knowledge necessary to create ontologies for specific domains, like medical ontologies, or ontologies for supply chain management.

Kochhar et al. collect human input for decisions in the Freebase knowledge base that an automated process cannot decide [KMP10]. They make use of paid contractors and to a lesser degree of volunteers to act as judges for these non-automatable decisions. To increase the quality of the judgments, they cultivate long-standing relationships with their judges. In addition, they identify judges with id and profile. The authors describe these relationships as the main reason for the good quality of the judgments. Thus, their results are unlikely to be transferable to typical online communities with an anonymous character. Further, the judge decisions could potentially benefit from the use of an HRM.

Von Ahn uses games to motivate users to perform useful tasks [vA06]. For example, users in the ESP Game are randomly paired to create tags for an image and receive points whenever their tags match. Siorpaes and Hepp [SH08] use this principle to build ontologies from Wikipedia entries by categorizing entries as either classes or instances. Users receive points when they agree on the categorization. Again, these approaches are better suited for knowledge domains that are known to the general public and where data is already available, e.g., in form of Wikipedia entries. Next, the game of assigning Wikipedia entries to predefined categories and rewarding users based on answers by other users is essentially an HRM setting. Thus, scaling the rewards for agreement by means of an HRM could give way to better results with these games.

3.2. Creating Structured Knowledge with Consensus Builder

This section describes Consensus Builder, our web-based tool that allows for the collaborative creation of structured knowledge. For our experiments, we use two different versions of Consensus Builder: Consensus Builder 1.0 (CB1), released in 2007, and Consensus Builder 2.0 (CB2), released in 2009. Both versions overlap to a large degree. In the following, we first describe CB2 since it represents our latest development. The description considers the different aspects of the functionality of CB2. Afterwards, we describe how CB1's functionality differs regarding each of these aspects.

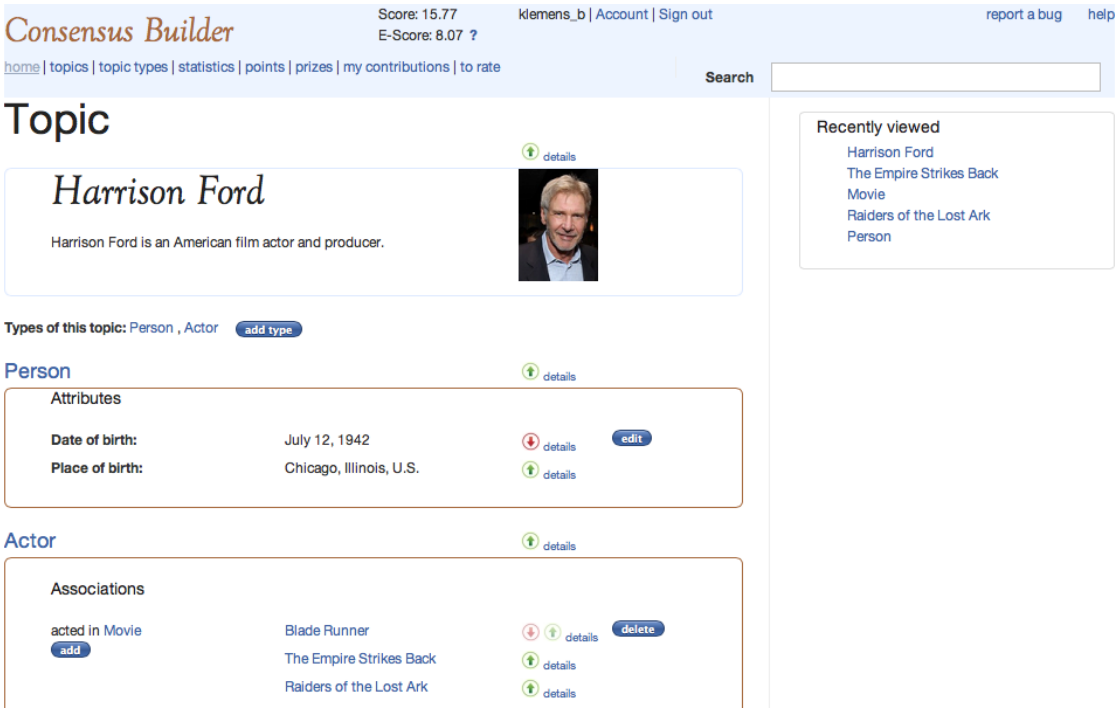


Figure 3.1.: Details for topic `Harrison Ford` in Consensus Builder 2.0

3.2.1. Consensus Builder 2.0

Data Format

CB2 uses a lightweight ontology data model similar to Topic Maps [Pep00] and Entity-Relationship Models [Che76] with some deviations for better usability. Data can be created on the type and on the instance level. That is, users can specify the schema and create instance data. Topics represent entities of the real world, e.g., `Harrison Ford` or `Indiana Jones`. Topics can have one or more types, e.g., `Harrison Ford` is of type `Person` and of type `Actor`. Types contain attributes and association types. Attributes describe simple data, like 'date of birth', and are constrained by data types, e.g., integer, string, or date. Association types describe associations between topic types, e.g., `Actor <acts in> Movie`.

Having said this, the objective of this chapter is the design and deployment of incentive mechanisms for the collaborative construction of structured knowledge. In this specific context, the data format to encode the knowledge is of secondary concern. We have mainly chosen the data format described above because of its ease of use for non-expert users. The functionality of Consensus Builder is applicable to other formats for structured knowledge as well, such as those specific to the Semantic Web. Furthermore, the data format currently used can be mapped to OWL in a straightforward way: Topic types are mapped to OWL classes, attributes to data-type properties, association types to object properties (as well as to their inverse properties) with domains and ranges restricted to

the respective classes or data types. Topics are mapped to instances, attribute values to literals, etc.

Collaborative Editing and Rating

Users can create and change all parts of the data model collaboratively. They can create single data elements like attributes or topics and change elements created by others. When a user adds a type to a topic, the topic inherits the attributes and association types of that type. For example, when a user adds the type `Actor` to the topic `Harrison Ford`, `Harrison Ford` inherits the association type `<acts in>` from `Actor`. Subsequently, users can set attribute values and add associations to the topic as specified by the added type. E.g., they can add topics of type `Movie` that `Harrison Ford` acted in, such as `Blade Runner`. CB2 takes care that users can create only attribute values and associations that are valid regarding the type level. In addition, CB2 provides various functions for browsing and searching, and contains functionality to discuss topics and topic types, and statistics such as user scores. Further, CB2 contains an *announcement* feature that allows to make announcements to the community of users or to subsets of the community.

Rating scheme. CB2 features a fine-grained association between contributions and ratings, called *rating scheme*, that lets users assess the quality of contributions on a very detailed level. Users can rate every element on the type as well as on the instance level that can be manipulated, e.g., topic names, attributes and association types, and attribute values.

A rating of a contribution x refers to the perceived correctness of x. The specific semantics differ depending on the contribution rated. For example, rating a type-instance relationship between type `Person` and topic `Harrison Ford` means evaluating the correctness of the statement "Harrison Ford is a Person". Rating the attribute `Date of birth` of type `Person` evaluates the correctness of the statement "Date of birth is an attribute of Person". The functions for rating, editing, and displaying the data are tightly integrated in the user interface; see Figure 3.1. (See also Figure A.1 in Appendix A.)

Rating scale. CB2 supports rating scales of arbitrary granularity. Since our rating scheme is very fine-grained, we prefer rating scales of lower granularity in order not to overburden the user. In general, we either use the well-known five-star rating scale or a binary rating scale ('low' vs. 'high'). Cosley et al. test scales of different granularity and find that ratings correlate strongly between scales, even though users give slightly higher mean ratings on the binary scale than on the more fine-grained scales. They conclude that "a designer might choose to allow users to rate on any scale they wish" [CLA+03].

Dealing with side effects of change operations. Users can change and delete individual contributions. However, change/delete operations are not trivial in any setting where data items depend on each other. For instance, what happens with other contributions and ratings associated with a contribution just deleted? Think of the deletion of a type that has associated topics and has received high ratings. To address these issues, we

operation	*points*(*operation*) in CB2	*points*(*operation*) in CB1
Creating a topic type	3	-
Creating a topic	3	3
Creating an association type	4	4
Creating an association	2	2
Creating an attribute	3	-
Creating an attribute value	2	2
Creating an occurrence	-	2
Adding a type to a topic	2	2

Table 3.1.: Number of points per operation in CB2 and CB1.

have made the following design decision. Users can only change/delete a contribution if it satisfies two conditions. First, it must not have dependent contributions. For example, an attribute can only be changed/deleted if there are no attribute values associated with it. Second, it either must not have received any ratings, or its average rating value must be below a certain threshold. Consequently, only contributions deemed low quality by the community can be changed/deleted. The user who has changed the contribution becomes its new owner.

Commission: Rating-Dependent Remuneration

To motivate users to create and maintain the data, we reward data operations with points based on ratings given by other users. We refer to the rating-based remunerations as *commission*. A user obtains a commission every time another user issues a favorable rating for a contribution that the first user is the owner of. (The owner of a given contribution x is the user who made the most recent change to x. If x has not been changed then x's creator is its owner.)

Let $r \in \{1, \ldots, m\}$ denote the value of a given rating for a given contribution. That is, in case of a five-star rating, r is the number of stars of that rating and $m = 5$. In case of binary ratings, $m = 2$, $r = 1$ denotes a 'low' rating, and $r = 2$ denotes a 'high' rating. We compute the commission in CB2 as

$$commission_{\text{CB2}} = \tau(r) \cdot points(operation), \tag{3.1}$$

where τ is a function depending on the scenario, and *points*(*operation*) depends on the operation. See Table 3.1 for an overview of *points*(*operation*). For instance, *points*(Create topic) = 3.0. (We specify the function τ used in the experiments in Section 3.5.5.)

Figure 3.2.: Screenshot of Consensus Builder 1.0: Details for topic `Relational Algebra`.

3.2.2. Differences of Consensus Builder 1.0 compared to Consensus Builder 2.0

Data Format

The data format of CB1 is closer to the Topic Maps format (cf. 'Topic Maps' on page 31). Here, as opposed to the model in CB2, a topic can have one or more types, e.g., the topic type `Professor` is of type `Instructor`. However, topic types do not constrain the attributes and associations that their instances can have. Instead, in CB1, association types partially fill the role of schema elements. Similar as in CB2, they constrain the types of the topics that can partake in the instantiated association. As opposed to CB2, association types are reified concepts that can exist on their own, i.e., independent of topic types. Further, CB1 supports occurrences, i.e., information sources that are relevant to a particular topic. A topic can have one or more occurrences that represent links to external web pages or documents, e.g., to the personal website of the professor.

Collaborative Editing and Rating

CB1's user interface for editing and rating is similar to that of CB2; see Figure 3.2 for a screenshot. (Further, see Figure A.2 in Appendix A.) However, since the type level does only partially constrain the instance level, users have more freedom when creating attributes, and occurrences for a topic in CB1. Here, only the creation of associations is constrained. One can interpret this as ratings of other users playing the role of the schema level: they indicate non-allowed attributes.

Similar to CB2, CB1 allows for deleting concepts that have no dependent concepts only. However, in CB1, a concept that has dependent concepts can be changed, for example an association type that has associated instances. In that case, CB1 invalidates all ratings associated with the changed concept.

Commission

CB1 pays commission *on top* of the points for the respective operation. That is, for each data operation the user receives a guaranteed reward of $points(operation)$ as described in Table 3.1. In addition to this guaranteed reward, a user receives a commission if another user rates his contributions favorably

$$commission_{\mathrm{CB1}} = \tau'(r), \tag{3.2}$$

where τ' is a function depending on the scenario. (We specify the function τ' used in the experiments in Section 3.5.5.)

3.3. Design Decisions Concerning the Honest Rating Mechanism

We want to elicit high-quality ratings, as opposed to ratings that are uninformed or simply copy the majority opinion. (This does not exclude the majority opinion from

after signal	$p(1)$	$p(2)$	$p(3)$	$p(4)$	$P(h)$
	0.25	0.25	0.25	0.25	0.5
l	0.40	0.30	0.20	0.10	0.4
h	0.20	0.30	0.30	0.20	0.5

Table 3.2.: Updating the type prior $p(t)$.

being correct.) However, a simple reward, e.g., one point per rating, does not suffice. It does not provide an incentive for the rater to gather information before issuing her rating and to respond truthfully. To address these challenges, we use an HRM, namely the Peer Prediction mechanism described in Section 2.2.1. In the following, we describe the design decisions behind our implementation of the HRM. For a detailed discussion of HRMs, see Section 2.2.

Estimating types and signal distributions. Let $Bin(n \mid q, N)$ denote a binomial distribution of getting n successes in N Bernoulli trials with success probability q. We encode types as $T = \{1, \ldots, \theta\}$. We assume that each type $t \in T$ generates a binomial distribution of signals with success probability $q = t/(\theta + 1)$. That is, type t generates the signal distribution $P(s_j \mid t) = Bin(j - 1 \mid t/(\theta + 1), m - 1)$.

For each contribution k, we maintain one local type distribution $p_k(t)$. We also maintain one global type distribution $p(t)$ that we use to initialize the local distribution of a newly created contribution. When a user submits a rating r for k, we update k's local distribution $p_k(t)$ as well as the global distribution $p(t)$ with r. Further, we assume that changing an item in the knowledge base also changes its quality and therefore renders previous ratings invalid. Consequently, if a user changes a contribution k, we reset k's rating history and initialize $p_k(t)$ with the global distribution at the time of change.

For an illustration of the process of updating $p(t)$, consider the following example adapted from [MRZ05], with two signals, 'low' ($s_1 = l$) and 'high' ($s_2 = h$), and four types, i.e., $T = \{1, 2, 3, 4\}$. Since there are only two signals, the binomial signal distribution degenerates to a Bernoulli distribution. That is, type t generates signal h with probability $P(h \mid t) = t/5$. In other words, types 1, 2, 3, and 4, generate h signals, 20, 40, 60, and 80 percent of the time, respectively. Table 3.2 shows the updating of $p(t)$ according to Bayes' theorem. Initially, $p(t)$ is uniform and the marginal probability for signal h, $P(h)$, is 0.5. After receiving rating l, the lower types become more likely (second table row). Subsequently, after receiving h (third row), $p(t)$ becomes symmetric again, but the extreme types (1 and 4) become less likely.

In general, spreading out the type space makes the changes of $p(t)$ smoother. In the experiments, we use types in the set $\{1, \ldots, 9\}$ because this resolution is sufficient for our purposes.

Sequential scoring. The original Peer Prediction mechanism scores all ratings for a given contribution simultaneously. That means, the HRM has to wait with scoring

until every user in the community has submitted her rating. Since this is impractical in many scenarios, the authors of the HRM suggest a *group-scoring extension* for sequential interaction [MRZ05]. That is, they suggest to put subsequent ratings into small groups of three or more ratings, and score ratings in the group simultaneously.[2] We adapt this extension for our purposes as follows. To score ratings as soon as possible, like the original group-scoring extension, we put subsequent ratings of a contribution into groups of small size (typically 3 or 4). However, instead of scoring ratings in a group only after the group is complete, we score each rating against the k-th next rating within that group. More formally, let g denote the group size. Suppose, there is a sequence of ratings $i = 1, 2, \ldots, I$ for a given contribution. We score each rating i against the reference rating

$$r(i) = ((i + k - 1) \bmod g) + 1 + \lfloor (i - 1)/g \rfloor \cdot g.$$

For example, for $g = 3$ and $k = 1$, the first tuples $(i, r(i))$ are $(1, 2), (2, 3), (3, 1), (4, 5)$, etc. This means, raters only have to wait for the next rater at most – or not wait at all, in case of rating $i = 3, 6, 9, \ldots$ – until their rating is scored. Thus, our adaptation scores raters more than 1.5 times earlier (given $g = 3$ and $k = 1$) than the original group-scoring extension, where raters have to wait for their score until their group is complete. Moreover, our adaptation also reduces the number of ratings that potentially remain unscored – because their rating group never got complete – compared to the original extension.

To guarantee the conditions for the Bayesian Nash equilibrium, we update the distributions $p_k(t)$ and $p(t)$ with $r(i)$ as soon as we have scored rating i. Further, users can view the mechanism's current estimate of $P(s)$ for a given contribution by clicking a 'details' link next to the contribution. To motivate the users that have unscored ratings, we display the sum of the expected scores for her unscored ratings.

Finally, we vary k and g randomly from time to time, so that raters cannot guess their reference rater easily. Thus, we hope to keep users from cheating, i.e., coordinating their ratings with their reference raters.

3.4. Hypotheses

Based on our research questions, we have formulated five hypotheses. They refer to

- the effects of ratings,
- the effects of commissions, and
- the effects of the HRM.

We use the template $\mathrm{H}^{\text{superscript}}_{\text{subscript}}$ for denoting a hypothesis. The superscript denotes the variable whose effects we want to study: 'rate' denotes ratings, 'comm' denotes

[2] Simply scoring each rating against the subsequent one is not feasible. In that case, the last rater has no incentive to tell the truth, and, consequently, the second last rater – who is scored against the last rater – has no incentive to tell the truth either, etc.

commissions, and 'hrm' denotes the HRM. The subscript denotes the effected variables: contribution quality ('cqual'), contribution quantity ('cquan'), rating quality ('rqual'). An exception to this template is the introductory hypothesis.

Having said this, our hypotheses are as follows.

$H_{\text{r-meas}}$: Ratings are a reliable measure of the quality of contributions. Rating mechanisms are always based on the assumption that individual ratings can be aggregated to a meaningful measure of quality. We suggest that ratings are indeed suitable to measure the quality of contributions.

$H^{\text{rate}}_{\text{cqual}}$: The presence of ratings increases the quality of contributions. Cosley et al. [CFK+05] (see also Section 2.1.4) have conducted a study in an online community where the contributions of a participant are assigned to another participant who can check the contribution for accuracy and correct it if necessary. Their results show that this checking improves the quality of contributions. We hypothesize that the same effect occurs when the checking/correcting of contributions is replaced by the broader concept of rating mechanisms. That is, we hypothesize that users create contributions of higher quality in Consensus Builder if they can rate each other's contributions.

$H^{\text{comm}}_{\text{cqual}}$: Commissions (both $\text{commission}_{\text{CB1}}$ and $\text{commission}_{\text{CB2}}$) increase the quality of contributions compared to static rewards. Commissions reward users for their contributions contingent on the ratings of their peers: the more favorable ratings a contributions receives the higher the reward. Static rewards, on the other hand, reward users for every contribution with a fixed number of points. The assumption behind commissions is that users put in more effort to create high-quality contributions if they are rewarded for quality. We expect his assumption to hold for both variants of commission, i.e., for $commission_{\text{CB1}}$ and for $commission_{\text{CB2}}$. This seems likely but is not self-evident.

$H^{\text{comm}}_{\text{cquan}}$: The fully rating-dependent $\text{commission}_{\text{CB2}}$ reduces the quantity of contributions compared to static rewards. In CB1, users receive a rating-dependent commission ($commission_{\text{CB1}}$) in addition to a fixed, i.e., rating-independent, reward per contribution (see Section 'Commission' on page 42). Based on our experience with CB1, we emphasized the rating-dependent part of commissions in CB2. That is, we removed the rating-independent reward and made $commission_{\text{CB2}}$ fully dependent on ratings (see Equation (3.1)). However, we expected a trade-off between quantity and quality: if users are exclusively rewarded contingent on quality they create contributions of higher quality (as the previous hypothesis states) but of lower quantity. Thus, prior to designing experiments for CB2, we formulated this hypothesis.

$H^{\text{hrm}}_{\text{rqual}}$: An HRM improves rating quality compared to static rewards. Hypothesis $H^{\text{rate}}_{\text{cqual}}$ above states that the presence of ratings has a positive impact on the quality of the created knowledge. $H^{\text{hrm}}_{\text{rqual}}$, on the other hand, is explicitly concerned with the quality of the ratings themselves. We expect that scoring ratings by means of the HRM motivates

raters to submit ratings of higher quality than a fixed score per rating. We deem such high-quality ratings essential for the creation of high-quality knowledge. This is because only high-quality ratings allow filtering out bad contributions and thereby increase the quality of the knowledge. Ratings of low or unknown quality cannot achieve this.

3.5. Experiments

To test our hypotheses, we conducted two series of experiments with Consensus Builder. The first series of experiments used CB1. In this first series, we wanted to gain empirical insights into how suitable Consensus Builder is as a tool for the collaborative creation of structured knowledge. The experiments of this series focus mainly on testing the hypotheses that regard the presence of ratings for the collaborative creation of structured knowledge. The second series of experiments used CB2. The respective experiments were primarily concerned with testing the hypothesis regarding the HRM. Both series tested the hypotheses concerned with the effects of commissions.

In the following, we first elaborate the design of the experiments. We then present the individual experiments of each series. Afterwards, we describe characteristics of the experimental setup common to all experiments, such as the configuration of the HRM and the commission functions, and external rewards participants received. Further, we discuss the different gold standards we used for quality assessment. Finally, we list the statistical methods we used for our analysis.

3.5.1. Experimental Design

To test our hypotheses, we conducted controlled single-blind experiments. That is, in each experiment, we randomly assigned participants to the experimental group (EG) or the control group (CG) without the participants' knowledge. We use the EG to evaluate the effects of the mechanisms in question. The CG serves as the baseline. For example, to test whether ratings affect contribution quality, we enabled ratings for the EG and disabled them for the CG. Such a controlled design allows us to establish cause and effect since the only variable that differs systematically between EG and CG is the respective experimental feature.

As independent variables (causes) we choose the following

- usage of ratings vs. no ratings,
- usage of commissions vs. static rewards, and
- HRM scoring of ratings vs. a static reward per rating.

As dependent variables (effects) we measure among others

- contribution quality,
- contribution quantity, and
- rating quality.

To measure rating and contribution quality, we employed different gold standards that we discuss below in Section 3.5.6.

Shared Data between Experimental Groups vs. Separate Data

Should EG and CG operate on the same data or should they use separate knowledge bases? This question refers to the internal validity of the experiment: Does the dependent variable measure the effect caused by the variation of the independent variable? Or does it measure some other undesired influence by a third variable? The answer to these questions depends on the interaction between the respective independent and the dependent variables, as we discuss in the following.

Separate data to measure contribution quality. To make sure that the differences in 'contribution quality' are indeed caused by variations of the independent variables 'usage of ratings' and 'usage of commission', both experimental groups have to operate on separate sets of data. To see why this is so, consider the alternative, i.e., using a shared knowledge base. In such a setting, it is impossible to differentiate between the effects of the independent variable and other undesired effects, such as the following three. (1) Data entries depend on each other, e.g., the contributions on the instance level depend on the schema level. That means, the quality of a contribution a can be influenced by the quality of another contribution b associated to a. And the former could have been created by a member of one group while the latter could have been created by a member of the other group. In that case, it is unclear what influenced the quality of a: The independent variable? The quality of b. A bit of both? (2) A contribution x could be created from a member of one group and afterwards be changed by a member of the other group. In that case, it is unclear which group to attribute the quality of x to. (3) When testing the effects of $\mathit{commission}_{\mathrm{CB2}}$, we expect a higher number of low quality contributions in the CG according to $\mathrm{H}^{\mathrm{comm}}_{\mathrm{cquan}}$ and $\mathrm{H}^{\mathrm{comm}}_{\mathrm{cqual}}$. This might affect the results even more if both groups operated on shared data. Only separate knowledge bases for each group let us eliminate such undesired effects on contribution quality and thus allow for an unambiguous assessment of the contribution quality of each group.

Shared data to measure rating quality. To measure the effects of the HRM on rating quality, on the other hand, both groups have to operate on the same data. If the groups used separate knowledge bases, it would be hard to tell whether differences in rating quality result from the usage of the HRM or from differences between the knowledge bases. For example, separate knowledge bases could differ regarding the average level of difficulty for assessing the data. In that case, the rating quality would not only reflect the effects of the HRM but also the effects caused by the variation in difficulty. Only when participants from both experimental groups rate items from the same set can such undesired effects on the dependent variable 'rating quality' be minimized.

3.5.2. Experiments Using Consensus Builder 1.0

We conducted the experiments with CB1 in the context of an existing online lecture community in January and February 2008. The community consisted of students attending the lectures of our department. The community portal of the lecture community contained tools for course management and collaboration (see Figure A.3 in Appendix A). For example, students could access the syllabus of the courses, check recent announcements, download lecture material, and use a discussion forum. In order to make CB1 available to the participants of the online community, we integrated CB1 into the community portal. To achieve a seamless integration, we facilitated navigation between the Topic Map created with CB1 and the other tools of the community. For example, users could link lecture slides as occurrences of topics. They could also rate the validity of these occurrences. In this lecture community setting, we conducted two experiments with CB1 that we describe in the following.

Experiment RATECOMPARE

For this experiment, we invited students of the lecture "Database Deployment" and instructed them to model the lecture as a Topic Map. At the beginning of the experiment, we gave a short introduction to Topic Maps and demonstrated our tool. We wanted to test H^{rate}_{cqual}, i.e., that the usage of rating mechanisms increases the quality of contributions. Thus, we enabled ratings for the EG and disabled ratings for the CG. In order to eliminate potential influences between the groups, we let each experimental group create its own Topic Map. The experiment lasted two weeks.

Experiment COMMISSION-CB1

For this experiment that we conducted subsequently to RATECOMPARE, we invited students of the lecture "Data Warehousing and Mining". The students of this lecture overlap almost completely with those of the lecture "Database Deployment" of RATECOMPARE. Consequently, we let already registered participants remain in their respective experimental group assigned to them in RATECOMPARE. Again, we instructed participants to model the lecture as a Topic Map and we gave general introductions to Topic Maps and Consensus Builder. To test H^{comm}_{cqual}, i.e., that commissions improve the quality of contributions, we chose *usage of commission* as the independent variable: users in the EG received the $commission_{CB1}$ while users in the CG did not receive a commission. Both EG and CG received a guaranteed reward for creating contributions as described in Table 3.1. We enabled ratings in the EG since they are the basis for commissions. To exclude the effect of 'ratings vs. no ratings' on contribution quality, we enabled ratings in the CG as well. Again, we let EG and CG work on separate Topic Maps to eliminate potential influences between the groups. The experiment lasted two weeks.

Testing $H_{r\text{-}meas}$

In each experiment conducted with CB1, at least one of the experimental groups used ratings. This allows us to test $H_{r\text{-}meas}$.

3.5.3. Experiments with Consensus Builder 2.0

We conducted a number of experiments with CB2 in March, July, and December 2010. Participants in the experiments with CB2 used Consensus Builder as a standalone tool. That is, as opposed to the CB1 experiments, we did not embed Consensus Builder into another community platform for these experiments.

Experiment Commission-cb2

We designed this experiment to test H_{cqual}^{comm} for $commission_{CB2}$ as well as H_{cquan}^{comm}. EG and CG operated on separate data in order to eliminate potential influences between the groups. For Commission-cb2, we invited students of the lecture "Database Systems". We asked participants to model the content of the lecture and also related information, and to rate each others' contributions. We rewarded ratings of both the EG and the CG by means of the HRM. The experiment lasted three weeks.

To test H_{cqual}^{comm} and H_{cquan}^{comm}, we chose *usage of commissions* as the independent variable. Users in the EG received the $commission_{CB2}$ while users in the CG received a rating-independent reward $points(operation)$ as defined in Table 3.1. To prevent potential exploitation, this amount was deducted when the contribution was deleted.

Experiment Rateonly

In this experiment, we focused exclusively on the HRM. To reduce effects that are not related to the HRM, we disabled all functionality for creating and editing contributions. This resulted in a modified CB2 which only supported rating, searching, and viewing. We recruited participants among students of our chair and instructed them to rate 127 contributions. We had preselected these 127 contributions from a knowledge base that students had created for the domain "Karlsruhe Institute of Technology" in a creation phase prior to the experiment itself. The selected contributions remained embedded within the knowledge base. That is, the participants of the experiment could see all the contributions from the creation phase, but could rate the 127 selected ones only. The experiment lasted three days. To test H_{rqual}^{hrm}, we scored participants in the EG with the HRM, while the CG was scored statically with one point per rating.

Experiment Honstudents

We tested H_{rqual}^{hrm} in a setting with the full functionality of CB2. We invited students of the lecture "Machine Design" of the Department of Mechanical Engineering of our university. We told them to create topics and types which represent the content of the lecture and to rate the contributions of others. Again, we rewarded the EG by means

of the HRM and the CG with one point per rating. To allow for comparing the rating quality of EG and CG later on, both groups worked together on the same knowledge base. The experiment lasted three weeks.

Experiment HONSTAFF

We repeated the experiment HONSTUDENTS to test H^{hrm}_{rqual} in a setting close to that of a community within a company. For this run, we invited researchers from the Institute of Product Engineering of our university. As knowledge domain we used a model for the engineering design process developed by this institute [ASM11]. We advised participants to use the elements of that engineering model as topic types and concrete instances as topics. The experiment lasted two weeks.

3.5.4. Real-World Significance of the Experimental Setups

Our experiments go well beyond vanilla laboratory experiments in several respects: They take place in real-world settings, within online communities where participants are not restricted by laboratory conditions. Unlike toy domains, the knowledge domains used were complex and had real-world significance. For example, the majority of the participants of the CB1 experiments stated that they planned to use the Topic Maps they had built for the preparation of their exams. The participants used Consensus Builder in an asynchronous fashion from home or from their workplace. We put attention to not letting participants know that an experiment was taking place nor that there existed different experimental groups. We achieved this by announcing the experiments as "beta test and user study". (In RATECOMPARE, some participants who knew each other in real life were assigned to different groups. When these participants asked why their acquaintances could rate contributions and they could not, we claimed that only few participants used rating mechanisms in order to calibrate system parameters.) During the CB1 experiments, we introduced experimental features to the members of the EG by means of email. During the CB2 experiments, we introduced them by means of the announcement feature within the Consensus Builder tool. Further, the assignment to groups was invisible to the participants, i.e., there were no indicators (e.g., specific URLs, etc.) that made the group explicit. The experiments lasted up to several weeks. This blurred the distinction between real world and experiment further. In general, participants remained anonymous to each other throughout the experiment and had no information about how many members their respective communities had. The university courses from which we recruited participants for the CB2 experiments had an anonymous character as well. All had a high number of students (up to 600).[3] Further, one of the lectures (experiment HONSTUDENTS) has been recorded and broadcast on the Internet, and most students chose to watch it from home. Thus, even though participants in these experiments were from the same course, we have not been aware of any personal interaction regarding the creation of structured knowledge (the exception being RATECOMPARE). This impression was confirmed in

[3] The lectures that supplied the participants for the CB1 experiments were smaller (about 60 students each) and therefore less anonymous.

personal discussions with participants who have come to our offices to collect their remunerations for participating. Thus, regarding the aspects that are relevant to the character of our study, the settings do not differ much from large online communities.

3.5.5. Further Characteristics of the Experiments.

Settings for Rating Scale and HRM Parameters

CB1. For the experiments with CB1, we used the well-known five-star rating scale, i.e., signals are members of the set $\{1, \ldots, 5\}$, where 1 represents the 'worst' and 5 the 'best possible' value. As HRM, we used the classical Peer Prediction mechanism with the quadratic scoring rule and scaled the remunerations for ratings to the interval $[0, 1]$.

CB2. For the experiments with CB2, we used a binary rating scale, i.e., ratings are either 'low' or 'high'. As mentioned in Section 3.2.1, users are not significantly influenced by the granularity of the rating scale [CLA+03]. Remember that in the CB2 experiments, we were primarily concerned with testing the hypothesis regarding the HRM. Therefore, we chose the binary rating scale since it makes conveying the intuition behind the HRM to users easier than more fine-granular scales. Accordingly, we have modeled signals to be in the set $\{l, h\}$. For CB2, we used the Peer Prediction HRM with the linear program defined in Equation (2.10). Therefore, we had to specify an upper bound for the external benefit a rater can gain from lying, i.e., reporting a different signal than she has received. Because in our experimental setups side payments for lying from third parties are unrealistic, we simply set the external lying benefits to a low value of $\lambda(s_j, s_l) = 0.5$ for all signals $s_l \neq s_j$. We set the external benefits gained from truth-telling to $\lambda(s_j, s_j) = 0$. (We avoid setting the external lying benefits to 0 because this would lead to the undesirable effect that raters receive the same payments for lying as for truth telling.) Further, we set the rating costs c to 1.5 points. Given the $\lambda(\cdot, \cdot)$ settings described above and $c = 1.5$, the linear program scales the expected payments for truth-telling to about 1.5 points, if the type prior is not very heavily skewed. This allows for a comparison with the CG, who was rewarded with 1 point per rating. We use 1.5 points instead of 1 for the EG because we assume risk-averse participants, as well as some unscored ratings.

Settings for Commission

We set $\tau' = 0.25(r-1)$. That is, the commission for CB1 experiments is $commission_{\text{CB1}} = 0.25(r-1)$. In other words, a user received 0.25 points for a two-star rating given to his/her contribution, 0.5 points for a three-star rating, and so on.

We set $\tau = 0.2(r-1)$. I.e., $commission_{\text{CB2}} = 0.2(r-1) \cdot points(operation)$. That is, the current owner of a contribution received $0.2 \cdot points(operation)$ for every positive rating the contribution had received and 0 points for negative ratings.

External Rewards

To recruit students for RATECOMPARE and COMMISSION-CB1, we announced to conduct a lottery of six exam bonuses. The intention behind the lottery was to motivate participants, including those with low scores. The high scores of other participants would likely deter those low-scoring participants if only the top-k members received a reward. The points gathered served as number of lots for the lottery. So the chance of winning an exam bonus was proportional to the final score in the experiment.

To motivate students to participate in COMMISSION-CB2 and HONSTUDENTS, each *active* participant received a guaranteed compensation of 5 Euro. A participant was considered active if he/she had reached at least 30 points. Additionally to the guaranteed compensation, we conducted a point-dependent lottery over $N_e \cdot 10$ Euro in total for each experiment e, where N_e denotes the number of active participants of e. The lottery consisted of $2 \cdot N_e$ draws of 5 Euro each. Every full point counted as an individual lottery ticket.

To motivate the researchers to participate in HONSTAFF we raffled off two digital cameras (Canon IXUS 105) and eight USB flash drives. Here, the chance of winning was proportional to the points gathered in the experiment as well.

For RATEONLY, we paid each participant of the CG a fixed compensation of 6 Euro. Each participant i of the N participants of the EG received $(3 + \frac{points(i)}{\sum_i points(i)} \cdot N) \cdot 7$ Euro, where $points(i)$ is the total number of points gathered by i during the experiment.

Note that, in an ideal scenario, external rewards scale in proportion to the points gathered. That means that the budget used for rewards would scale with the number of points as well. However, for our experiments, we only had a fixed amount of rewards at our disposal. The fixed amount might introduce some competition, as we discuss in Section 3.7.

Training Phase for CB2

We noticed that contribution quality increased between RATECOMPARE and COMMISSION-CB1 in both experimental groups. We attribute this partially to learning effects, since, as mentioned, the participants of RATECOMPARE and COMMISSION-CB1 overlapped to a large degree. To mitigate such potential learning effects in the CB2 experiments, we added a training phase prior to each of these experiment. During this training phase, participants got accustomed with the Consensus Builder tool. To aid learning, we provided screencasts that explained the usage of CB and the details of data modeling in particular.

(Late) Registration

Anonymous accounts raise the problem of Sybil attacks [Dou02] where users forge multiple identities to gain larger influence. To counter those attacks in the CB1 experiments, we applied validation techniques to make sure that each person created only one account. For this purpose, we distributed activation keys among the students and manually ensured

that each person obtained at most one key. This activation key was prompted during the creation of an account and synchronized with the database. In the experiments with CB2, the domain of the email address constrained the registration to guard against Sybil attacks.

Participants could enter a CB2 experiment while it was already running. An algorithm based on biased coin randomization [Efr71] assigned participants to either the EG or the CG, while keeping the numbers of members of the groups balanced.

Questionnaire

After each experiment, we invited the participants to complete a questionnaire. It elicited feedback on rewards, ratings, rating mechanisms, the behavior of other participants, and the usability of the Consensus Builder tool. The number of questions per questionnaire ranged up to 30, dependent on the configuration for the respective participant. We used a five point Likert scale response format ('strongly disagree' to 'strongly agree') for most questions. To motivate participants to fill out the questionnaire, they received extra points upon successful completion.

3.5.6. Gold Standards for Quality Assessment

In the following, we discuss the different gold standards we used to assess the quality of contributions and ratings.

Expert Ratings

For the CB1 experiments, as well as for Commission-cb2, Honstudents, and Honstaff, we let domain experts rate a subset of contributions and used their ratings as gold standard. The subset of contributions, as well as the rating scale experts used, depended on the hypothesis to test. Testing H^{hrm}_{rqual} required comparing the rating quality between the EG and the CG. We randomly picked 150 contributions that had received at least one rating from both experimental groups. Here, experts used the same rating scale as users. To test H^{comm}_{cqual} and H^{rate}_{cqual}, we simply picked 50 contributions randomly from each group. Here, experts used a five-star ratings scale to allow for a fine-granular assessment of the contribution quality. For Commission-cb2, in addition to the 50 randomly selected contributions such as attributes, associations etc., we let the experts assess the 'overall quality' and 'overall adequateness' of the topic or topic type associated with the respective contribution as a whole. This allows for a comprehensive quality assessment of the contributions. In summary, apart from Commission-cb2, the experts used the same user interface as the participants to issue ratings. To understand the context, experts could see all contributions created in the respective experiment. To remove potential bias, experts had no knowledge about which experimental group the contributions they rated belonged to.

We chose the following domain experts for the various experiments: up to three teaching assistants (numbers in brackets) of the respective course for Ratecompare (three), Commission-cb1 (three), Commission-cb2 (two), and Honstudents (one).

For HONSTAFF, we chose a scientist whose research topic is the engineering model that served as the domain for the experiment. Since each of the domain experts had limited experience in data modeling for the latter two experiments, a database expert supported them with the assessment of the data model.

Inter-rater agreement between experts. For each of the experiments RATECOMPARE, COMMISSION-CB1, and COMMISSION-CB2 we let multiple experts rate the contributions. To make sure that our experts were in agreement, we computed a pairwise Cohen's kappa measure for experts that rated the same sets of data. Cohen's kappa is a statistical measure of inter-rater agreement [FLP13]. A kappa value of less than 0 stands for less than chance agreement, while a value of 1 means perfect agreement. Since the expert ratings are ordinal data, we computed the weighted kappa [Coh68] with square weights (disagreements are weighted according to their squared distance from perfect agreement). The computed kappa values were all statistically significant ($p < 0.01$) and ranged from 0.21 to 0.61 ($mean = 0.42$, $se = 0.035$), indicating fair to substantial agreement [LK$^+$77]. After we had computed the kappa values, we let each expert group discuss the controversial ratings to find a final consensus. We use these consensus ratings for the statistical analysis below.

As mentioned, for the domains modeled for HONSTUDENTS and HONSTAFF we had to recruit experts from another department. Here, we could only mobilize one domain expert for each experiment. However, we deemed the situation of using only one domain expert as acceptable since our previous experiments had shown that the opinions of multiple experts overlap to a sufficiently large degree.

Manipulation

For RATEONLY, we selected 127 contributions manually (all on the instance level) out of the more than 5000 contributions created during the data-creation phase. The contributions selected were unambiguously correct, as confirmed by information publicly available on websites. We manipulated 34 out of the 127 contributions so that they were false. The manipulated contributions together with the remaining manually selected ones served as the gold standard.

We classified the manipulations in three categories according to the effort needed to verify the respective errors:

1. *Easy to verify.* These are blatant errors, like a building having 666 elevators or a paper on sensor networks published in 1920.

2. *Medium effort to verify.* This category contained plausible-looking errors, like changes in room numbers or changes in co-authors of a paper. They could be detected by internet search.

3. *Hard to verify.* These manipulations were subtle and could only be verified with high effort, for example, the number of floors in a remote building.

We hypothesized that the HRM has an effect on errors of Category 2 only. Both groups should recognize errors in Category 1. Category 1 allows checking whether participants made any effort at all. For Category 3, the effort for error detection exceeded the benefit from honest ratings by much. It served as an extra check to exclude the possibility that the EG was more motivated than the CG a priori.

3.5.7. Overview Experimental Setups

Table 3.3 shows an overview of the different setups for CB1 and CB2 experiments: Columns contain the experiments. Rows list the respective characteristics.

3.5.8. Statistical Methods

We test the correlation between aggregated user and expert ratings with Spearman's rank correlation coefficient ρ. Further, we use Spearman's ρ to test the correlation between Likert responses from the questionnaire and other experimental results. We use Pearson's χ^2 test to evaluate associations between binary variables, e.g., between classification errors and usage of the HRM. (For directional associations we use the one-tailed χ^2 test [FLP13].) We use Welch's t-test to evaluate the difference of the mean number of contributions. We compare the five-star expert ratings by means of the Mann-Whitney U test [dWD10]. We carry out the analysis by means of the statistical software R [R C13].

3.6. Results

We present the results of our experiments, including the evaluation of the hypotheses and of the questionnaire. Table 3.4 gives an overview of the number of participants, contributions, and ratings for each of the six experiments.

3.6.1. $H_{r\text{-}meas}$: Ratings are a Reliable Measure of Contribution Quality

Many online communities aggregate ratings and display the average value as a measure of contribution quality. In order to analyze whether aggregated ratings are suitable to measure the quality of contributions, we compared user ratings gathered in experiments RATECOMPARE and COMMISSION-CB1 with expert ratings for the same contributions. Since the ratings of each contribution are averaged, as many ratings as possible are required to generalize from the subjective opinions of individual members. To have at least as many user ratings as expert ratings, we considered only contributions with at least three ratings from different users. An analysis of the data shows that there is a medium correlation ($\rho = 0.46, p < 0.01$) between the average ratings of users and experts. Because of the significant correlation between user and expert ratings, we deem the first hypothesis supported.

It is worth mentioning that the average of the expert ratings ($mean = 0.56$, $se = 0.03$; ratings normalized to zero-to-one range) is lower than the average of the user ratings

	CB1		CB2			
	RATECOMPARE	COMMISSION-CB1	COMMISSION-CB2	RATEONLY	HONSTUDENTS	HONSTAFF
Designed for	H^{rate}_{cqual}	H^{comm}_{cqual}	H^{comm}_{cqual},H^{comm}_{cquan}	H^{hrm}_{rqual}	H^{hrm}_{rqual}	H^{hrm}_{rqual}
Tests also	$H_{r\text{-}meas}$	$H_{r\text{-}meas}$	–	–	–	–
Rating Scale	5-star (EG only)	5-star	binary	binary	binary	binary
Static Ratings	–	–	–	CG	CG	CG
HRM	EG	both	both	EG	EG	EG
Static Contrib.	both	both	CG	–	–	–
Commission	–	EG	EG	both	both	both
Shared Data	no	no	no	yes	yes	yes
Gold Standard	Experts	Experts	Experts	Manipulation	Experts	Experts
Duration	2 weeks	2 weeks	3 weeks	3 days	3 weeks	2 weeks
Date	Feb. 2008	Feb. 2008	July 2010	Mar. 2010	July 2010	Dec. 2010

Table 3.3.: Overview of the experimental setups.

	CB1				CB2							
	RATECOMPARE		COMMISSION-CB1		COMMISSION-CB2		RATEONLY*		HONSTUDENTS		HONSTAFF	
	CG	EG	CG	EG	CG	EG	CG	EG	CG	EG	CG	EG
Participants	12	11	15	17	11	14	3	6	8	12	10	10
Contributions	1111	720	565	1113	802	206	127	127	151	1052	136	162
Ratings	–	119	1105	1699	180	151	381	762	943	456	419	555
Ratings/Contribs.	–	0.17	1.96	1.53	0.22	0.73	3	6	0.78	0.38	1.4	1.86

Table 3.4.: Summary of participation in the experiments. (*Number of contributions and ratings fixed.)

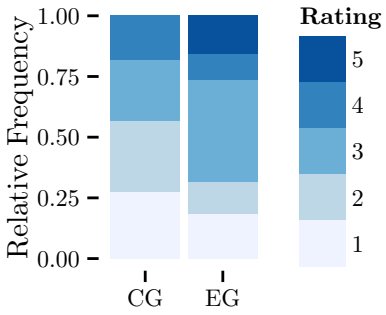


Figure 3.3.: Distribution of expert ratings for contributions in RATECOMPARE. (5 is the best-possible rating, and 1 the worst-possible rating a contribution could receive.)

($mean = 0.66$, $se = 0.03$). In other words, the experts were more critical than the users in their assessment of contribution quality.

3.6.2. H^{rate}_{cqual}: The Presence of Ratings Improves the Quality of Contributions

To test if the presence of ratings improves contribution quality, we use the data gathered in RATECOMPARE. Remember that we enabled ratings for the EG and disabled ratings for the CG in this experiment. As Figure 3.3 shows, the experts rated the contributions by the EG more favorably – even though only to a moderate extent – than those by the CG. The difference is statistically significant (Mann-Whithney U = 626, $p < 0.05$). Thus we deem hypothesis H^{rate}_{cqual} supported.

3.6.3. H^{comm}_{cqual}: Commissions (both $commission_{CB1}$ and $commission_{CB2}$) Increase Contribution Quality

For H^{comm}_{cqual} we distinguish between the commission paid additionally to guaranteed rewards ($commission_{CB1}$) and commission paid for favorable ratings only ($commission_{CB2}$).

In experiment COMMISSION-CB1, the CG received a fixed reward per contribution while the EG received points according to $commission_{CB1}$ additionally to the fixed reward. We find no statistically significant difference between the distributions of the expert ratings for the EG and the CG (Mann-Whithney U = 1723.5, $p = 0.8$). Even though the EG earned more five-star ratings from the experts, it received less four-star ratings than the CG (Figure 3.4). The remaining ratings (three-, two-, one-star) have about the same relative frequency in both groups.

In experiment COMMISSION-CB2, the EG received points according to $commission_{CB2}$ while the CG received a fixed reward per contribution. Figure 3.5 shows the distributions of expert ratings for individual contributions, as well as for 'overall quality', and 'overall adequateness' of the respective topics and topic types. We find statistically significant differences between the ratings for CG and EG in all three expert rating categories. That

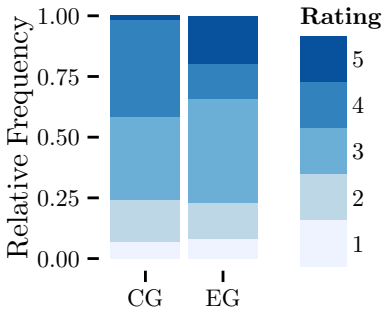


Figure 3.4.: Distribution of expert ratings in Commission-cb1. (5: best-possible, 1: worst-possible rating.)

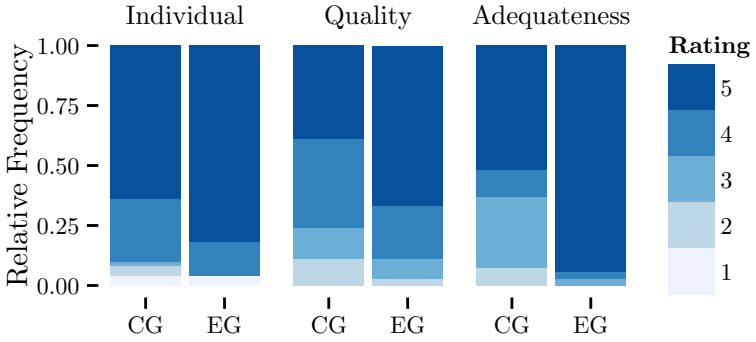


Figure 3.5.: Commission-cb2: distributions of expert ratings for individual contributions, as well as for 'overall quality', and 'overall adequateness'. (5: best-possible, 1: worst-possible rating.)

is, in each category, experts assigned significantly lower ratings to the CG (rewarded statically) than to the EG (rewarded with $commission_{\text{CB2}}$): (1) individual contributions (Mann-Whithney U = 1023.5, $p < 0.05$), (2) 'overall quality' (Mann-Whithney U = 684, $p < 0.05$), (3) 'overall adequateness' (Mann-Whithney U = 549, $p < 0.01$).

According to these results, we deem hypothesis $H^{\text{comm}}_{\text{cqual}}$ as partially supported. It is supported for $commission_{\text{CB2}}$, but not for $commission_{\text{CB1}}$.

Interestingly, we find for Commission-cb1 that members of the EG were significantly more critical with their ratings ($mean = 0.59$, $se = 0.39$; ratings normalized to zero-to-one range) than members of the CG ($mean = 0.72$, $se = 0.32$)(Welch's $t(2655.8) = 9.63$, $p < 0.01$). This fact conflicts with the assessment by the experts who evaluated contributions by both groups as of about equal quality. Apparently, the EG was biased towards more critical ratings and displayed a more competitive behavior compared to the CG. The same result holds true for Commission-cb2 as well. Here, participants remunerated with $commission_{\text{CB2}}$ seem to rate their peers' contributions more critically, too. The ratio of negative ratings was significantly higher in the EG (0.258) than in the CG (0.039) ($\chi^2(1)$=31.2, $p < 0.01$), even though the experts rated the contributions of

the EG more favorably. We discuss the implications of this finding in Section 3.7.

3.6.4. H^{comm}_{cquan}: commission$_{CB2}$ Reduces Contribution Quantity

We tested H^{comm}_{cquan} in Commission-cb2. There were much fewer contributions per participant in the group using $commission_{CB2}$ ($mean = 27.0$, $median = 22$ $se = 7.96$) than in the one without ($mean = 126.5$, $median = 46.5$, $se = 67.27$). However, there was a lot of variation between the number of contributions of individual participants as is reflected by the high standard errors of the means. Further, the difference of the means is not statistically significant (one-tailed Welch's $t(5.14) = 1.47$, $p = 0.10$). Consequently, even though the numbers are suggestive, we deem H^{comm}_{cquan} as not supported.

3.6.5. H^{hrm}_{rqual}: An HRM Improves Rating Quality

To test H^{hrm}_{rqual}, we compare the error rates of ratings rewarded with the HRM (EG) to those rewarded statically (CG). Let $r \in \{0, 1\}$ denote a given rating, and let $g(r) \in \{0, 1\}$ denote the gold standard of that rating.[4] We define the rating error $re(r)$ as an indicator variable

$$re(r) = \begin{cases} 1 & \text{if } r \neq g(r) \\ 0 & \text{otherwise.} \end{cases}$$

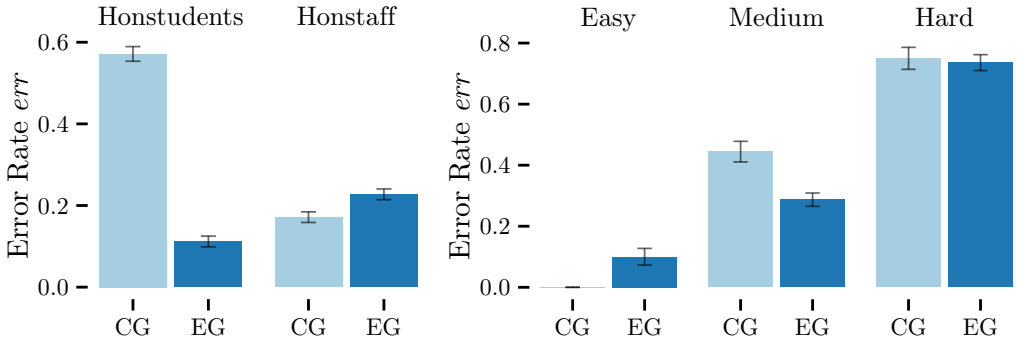
The *error rate of ratings (err)* is the proportion of rating errors out of all ratings, i.e., $err = \frac{1}{|G|} \sum_{r \in G} re(r)$, where G is the set of ratings which we have a gold standard for.

For Honstudents (see Figure 3.6a), there was a highly significant association between rating errors and the usage of the HRM ($\chi^2(1) = 71.52$, $p < 0.01$). The *err* was much higher for the CG (0.57) than for the EG (0.11). The odds ratio of making a rating error when using the HRM was 0.09. We conclude for this experiment that the mechanism improved rating quality.

For Honstaff (see Figure 3.6a) we found no statistically significant association between rating errors and the usage of the HRM ($\chi^2(1) = 1.9071$, $p = 0.17$). The CG showed slightly better results regarding rating quality (err=0.16) than the EG (err=0.22). For Honstaff, we conclude that there is no significant effect of the HRM on rating quality. A possible reason for this is that the researchers already had a high intrinsic motivation to create high-quality data since they wanted to use it in their research later on. Further, even though they did not interact outside of CB for the knowledge-creation task, they might have felt stronger obligations towards their relatively close-knit group. This high intrinsic motivation might have diminished the effects of the HRM.

Figure 3.6b shows the *err* for the three error categories of Rateonly for CG and EG, respectively. The participants of the EG made significantly fewer errors in the 'medium effort to verify' category (odds ratio = 0.5, $\chi^2(1) = 3.3$, one-tailed $p < 0.05$). For the other two error categories, we found no significant association between errors and the usage of the HRM('easy to verify': $\chi^2(1) = 1.52$, $p = 0.22$; 'hard to verify': $\chi^2(1) = 0.002$, $p = 0.96$). The *err* for ratings of non-manipulated contributions was very low in both

[4]The gold standard of a rating simply is the gold standard of the contribution the rating belongs to.



(a) Error rates for HONSTUDENTS and HONSTAFF.

(b) Error rates by error category for RATEONLY.

Figure 3.6.: Error rate of ratings for CG and EG.

Hypothesis	Result
$H_{\text{r-meas}}$	supported
$H^{\text{rate}}_{\text{cqual}}$	supported
$H^{\text{comm}}_{\text{cqual}}$ for $commission_{\text{CB1}}$	not supported
$H^{\text{comm}}_{\text{cqual}}$ for $commission_{\text{CB2}}$	supported
$H^{\text{comm}}_{\text{cquan}}$	not supported (but suggestive)
$H^{\text{hrm}}_{\text{rqual}}$	supported in two out of three experiments

Table 3.5.: Overview results regarding hypotheses.

groups (CG: 0.054, EG: 0.043) and the association not statistically significant (two-tailed $\chi^2(1) = 0.27$, $p = 0.6$). We conclude that, for RATEONLY, the HRM increases rating quality.

Summing up, we find that two out of three experiments support $H^{\text{hrm}}_{\text{rqual}}$.

Note that the low error rate for HRM ratings also supports $H_{\text{r-meas}}$, i.e., ratings are a reliable measure of contribution quality. This is because the less rating errors occur, the more correct and thus reliable the ratings are as a measure of contribution quality. In other words, the results obtained for $H_{\text{r-meas}}$ show that aggregated ratings are a reliable measure of contribution quality. The results obtained for $H^{\text{hrm}}_{\text{rqual}}$ show that this is more likely to be true for HRM-scored ratings than for statically rewarded ones.

3.6.6. Summary of the Hypotheses Tests

The empirical results we obtained support two of our hypotheses fully and two partially. Table 3.5 shows an overview.

3.6.7. Evaluation of the Questionnaire

19 participants of the CB1 experiments, and 27 participants of the CB2 experiments answered the respective questionnaire. The questions differed somewhat between the questionnaires used for CB1 and CB2. Further, the number of questions differed between experimental groups because the different experimental conditions rendered some questions meaningless for the respective groups. (E.g., we did not ask participants whose ratings we had scored statically whether they had understood the HRM.) Figure 3.7 shows an overview of the results for selected questions.

In the following we analyze the answers to the respective questionnaires in more detail.

Analysis of the CB1 questionnaire responses. Remember that at the end of the CB1 experiments, we raffled off exam bonuses as the sole external reward to motivate participation. In the questionnaire, we asked participants if they would have preferred deterministic rewards for the top k users, and further, if they considered the lottery of rewards as fair. Somewhat expected, user score and the desire for a deterministic rewards were correlated ($\rho = 0.55$, $p < .05$), while user score and viewing the lottery as fair were anti-correlated ($\rho = -0.47$, $p < 0.05$). In other words, participants with higher scores deem the lottery less fair and show a stronger desire for deterministic rewards. We found a moderate negative correlation between rating truthfully and joining the consensus rating ($\rho = -0.48$, $p < .05$). That is, the participants apparently consider truthful and consensus-oriented ratings as mutually exclusive. Further, the EG in Commission-cb1 reported more often that they tried to keep the scores of their fellow members low than the CG, although the difference was not statistically significant (Mann-Whithney U = 28.5, $p = 0.14$). This confirms our observation that there is a conflict between truthful ratings and commissions for good contributions. We discuss possible implications in the next section.

Analysis of CB2 questionnaire responses. We asked participants which rating strategy they used to maximize their rating score. Some stated an altruistic attitude "I did not intend to get as many points as possible, but tried to increase the quality of contributions by rating pointless or bad contributions as bad.", "I tried to rate as much as possible as honestly as possible." (both rewarded by the HRM). Others said they tried to maximize their scores, although with different rating strategies, dependent on their respective scoring mechanism, namely "Rating many items, but only those whose quality was easy to decide." (HRM), and "Simply rated everything, no matter how." (static reward for ratings).

Further, we analyzed the correlations of experimental results of the participants and their questionnaire answers. Not surprisingly, we find a positive correlation between the understanding of the HRM and the number of ratings ($\rho = 0.52$, $p < .05$). There is a strong correlation between the number of ratings and the stated strategy of rating the contributions of others badly in order to keep their scores low. However, this holds true only for raters whose ratings were scored statically ($\rho = 0.764$, $p < 0.5$). For participants using the HRM, on the other hand, this correlation was negligible ($\rho = 0.16$, $p = 0.59$).

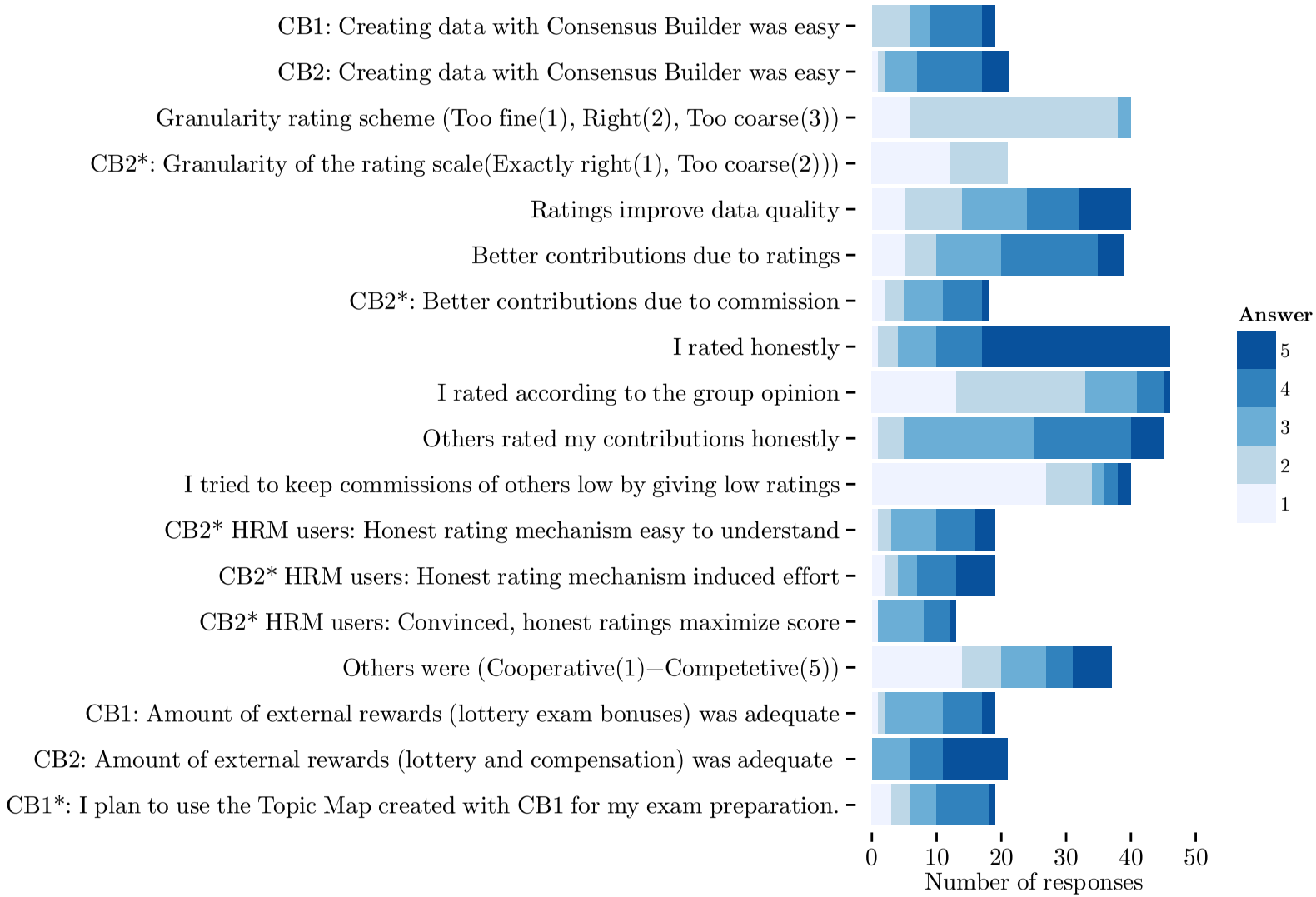


Figure 3.7.: Number of responses for selected questionnaire questions for CB1 and CB2 experiments. Answers range from 1: 'strongly disagree' to 5: 'strongly agree', unless otherwise noted. (CB1*/CB2*: question was part of CB1/CB2 questionnaire only.)

We find a weak correlation between the understanding of the HRM and the score received per rating ($\rho = 0.31$, $p = .18$). In other words, it is not necessary to understand the mechanism in order to profit from it.

3.7. Discussion and Lessons Learned

We discuss the experimental results and present the lessons learned while conducting the experiment.

More critical ratings due to commissions. Both forms of commission we tested – $commission_{\text{CB1}}$ and $commission_{\text{CB2}}$ – lead to somewhat more competitive user behavior: Users rewarded with commission give more critical ratings than users not rewarded with commission. Further, the responses to the questionnaire for CB1 confirm the more competitive behavior. We speculate that this is due to the zero-sum character of the external lottery rewards we offered. I.e., participants competed for a fixed number of lottery prices. And since positive ratings for other participants' contributions increase the others' chances of winning said price, users might rate more negatively. If our speculation is correct, a simple solution to this problem is to remove the zero-sum character by remunerating users not with a fixed number of rewards but with rewards proportional to the number of points. Of course, this might drastically increase the budget that is required to pay for commissions and the HRM scores, and thus is not an option for most online communities. Further, many online communities motivate users by prestige or status within the community, which is by definition zero-sum, at least concerning the overall status. Subdividing the overall status hierarchy into many might alleviate the zero-sum character. In that case, users can compete to be the best in many different categories. However, even though our data indicate somewhat more critical ratings due to commissions, the majority of ratings was of high quality. This is also supported by the results gained in HONSTUDENTS and HONSTAFF, which both show a high rating quality, at least for the participants rewarded by the HRM.

Besides the truth-telling Nash-Equilibrium, lying equilibria are also possible for the HRM mechanism (Section 2.2.3). In a lying equilibrium, raters agree on rating all contributions or a subset of contributions as either good or bad. Such undesired equilbria might be facilitated by commissions in zero-sum conditions: In order to avoid giving points to possible competitors, it might be conceivable that users – tacitly or explicitly – agree on rating all contributions as bad, while still being rewarded by the HRM. We did not find any evidence of a lying equilibrium.

Intrinsic vs. extrinsic motivation. Participants have been intrinsically motivated to some degree. They made good contributions and gave high-quality ratings even when they did not receive an extra reward for it. However, one of our results is that contributions and ratings are at least as good or better in the presence of commissions and the HRM, respectively. We speculate that, at least to some degree, the intrinsic motivation resulted from the fixed monetary compensation for participation, insofar as participants felt

they had to offer at least some effort. In fact, when planning the experiments, given a fixed total budget, we were confronted with a tradeoff between two quantities: On the one hand, the guaranteed compensation for participating. On the other hand, the score-dependent external rewards for commissions and rating scores. A low guaranteed compensation results in fewer participants. A high guaranteed compensation provides less incentive from rating dependent rewards. This is a known problem in the literature. For example, a study about the effects of monetary incentives on survey responses finds that larger monetary incentives produce (1) a higher response rate, (2) a greater degree of effort expended, and (3) more favorable comments towards the survey sponsor [JB90]. Therefore, the results gained in favor of the HRM and of commission might have been even stronger, had the respective CGs not been rewarded by means of a fixed reward.

Ratings sparsity. Despite the rewards offered, ratings are sparse in both CG and EG in the experiments with a variable number of ratings. That is, there are not nearly as many ratings per contribution as there are participants per group (see Table 3.4). Several reasons are possible: (1) Some participants do not like to rate data items they do not understand well enough even if they receive a guaranteed (static) score. For example, one participant of RATEONLY dropped out after the creation phase because she did not feel sufficiently familiar with the knowledge domain. (2) The number of points for ratings was too low compared to the *points*(*operation*) for contributions (cf. Table 3.1 and 'Settings for Rating Scale and HRM Parameters' on page 51). In that case, increasing the number of points per rating would solve the problem. (3) Many ratings remained unscored which might deter users from rating. For example, 65 percent of ratings by the EG in the experiment COMISSION-CB2 remained unscored by the HRM. This is because the HRM only scores a rating if there is a reference rating available, as described under 'Sequential scoring' on page 43 in Section 3.3. This might be a chicken-and-egg problem: If only a few ratings are scored, users rate less because they might get the impression that ratings are not rewarded as expected. But if users rate less, there are fewer reference ratings and thus only few ratings get scored. Higher expected points per rating and longer running experiments might solve this problem. (4) The guaranteed compensation for participation, as discussed above, might be another reason for the low number of ratings.

In future work, instead of fixing the points per contribution and the expected score per rating a priori, a mechanism could dynamically adjust them according to their relative shortage. For example, if ratings are scarce in a community, the expected score for ratings could be scaled to attract more ratings. The mechanism could be further refined to assign higher rewards for areas in the knowledge base, where contributions or ratings are particularly sparse.

Fake HRM vs. the real one. The results show a weak correlation between rating scores and understanding of the mechanism. An interesting question is whether a fake HRM would have the same effects as the real one. We speculate that telling participants that an HRM is used while scoring with some fake mechanism (for example, randomly) would still yield comparable results, at least in the short run. This could be tested experimentally

by comparing the alternatives 'no mechanism', 'real HRM', and 'fake HRM'. Note that, even if a fake mechanism yielded results similar to those of the real mechanism, the real mechanism would still be at least as good (or better in case participants realized the fake).

Evidence that a fake mechanism could work in the short run comes from Shaw et al. [SHC11]. They find that telling crowd workers the idea behind the Bayesian Truth Serum HRM ("You will receive a bonus payment if your answer is more common than collectively predicted."), without performing the actual computation of the score, is sufficient to incentivize the workers to answer survey questions more accurately. Their task was rather short lived, so the same reasoning as above applies: In the long run, the fake would likely be discovered.

Appropriate domains for experiments are hard to find. It turned out to be difficult to find domains with all of the following characteristics: (a) They are sufficiently controversial to generate variance in the ratings. (b) The experimenters, but not the participants, have access to the gold standard. For example, in the creation phase before the RATEONLY experiment, participants kept the schema extremely simple and almost exclusively copied data publicly available on the web. Since the contributions were almost completely correct, there were no negative ratings and hence no variance in the rating values. This means that we could not have measured the effects of our mechanisms on either contribution or rating quality of the creation phase meaningfully.

Surprisingly good data modeling. The quality of the schema created by participants not familiar with data modeling was surprisingly good. Despite some beginner mistakes (confusion of normal associations and type associations, creation of topic type 'Properties') the quality of the schema level was good and detailed.

3.7.1. Design Recommendations

Putting the labor intensive task of creating and maintaining structured knowledge into the hands of online communities is a feasible approach. The results of our study can give recommendations for the design of such communities. In general, if the intrinsic motivation is not sufficient for motivating users to contribute to an online community, extrinsic incentives can be provided. Such incentives are important to tie committed and thus valuable members to the community. Our experimental study confirms that rewards are a potential means to motivate users to contribute to an online community and create high-quality data.

In particular, the following design recommendations might be of help:

- *Ratings are better than no ratings.* They measure the quality of contributions reliably and increase the quality of the knowledge.
- *Allow for a fine-grained association of ratings to contributions.* We have not tested this per hypothesis. However, our experience gained in the experiments described above, as well as in some beta-tests with Consensus Builder, suggest that the

fine-grained ratings scheme is helpful for a detailed assessment of contribution quality. Further, the overwhelming majority of the questionnaire responses assess the granularity of the rating scheme as 'exactly right' Figure 3.7.

- *Use an HRM.* Users react to the presence of an HRM with ratings that are at least as good or better than statically rewarded ratings.
- *Use fully rating-dependent commissions.* Rewarding users for good contributions based on the positive ratings of their peers is beneficial. However, the rewards must be fully rather than only partially rating dependent, as our results show.
- *Expect somewhat increased competitiveness due to rating-dependent rewards*, in particular in zero-sum games for fixed prices or prestige within the community.

3.8. Conclusion

In this chapter, we have investigated peer production as a means to create structured knowledge collaboratively. We have proposed the usage of ratings and rating-based incentive mechanisms to stimulate contributions and to increase the quality of the knowledge created. In particular, we have discussed how a mechanism for honest ratings can be applied to this scenario. To evaluate our approach, we have developed a platform for the collaborative creation of structured knowledge called Consensus Builder. We have formulated hypotheses and designed experiments to test the effects of reward mechanisms on the quality of contributions to the structured knowledge base as well as on the quality of ratings. Using Consensus Builder, we have carried out six experiments that took place within different online communities. The online communities constructed complex knowledge domains in settings close to real-world scenarios. The results of the experimental study show that our platform is suitable for the collaborative creation of structured knowledge. Ratings prove to be a reliable measure for the quality of contributions in an online community. Further, we find that the usage of rating mechanisms, as well as fully rating-dependent rewards for good contributions, increase the quality of contributions. Finally, we find that an honest rating mechanism improves the quality of ratings in two out of three experiments.

4. On the Accuracy of Classification Schemes for Contributions in Peer-Rating Online Communities

In the previous chapter we have focused on measuring and improving the quality of online-community contributions. Now, we turn to the problem of how to classify contributions, i.e., how to decide to which class of a set of predefined classes a contribution belongs. Specifically, we investigate how to classify contributions based on the ratings they have received. This is relevant for the increasing number of online communities whose contributions serve as input for automatic processing. As an example, consider an online community that creates an ontology. Here, the community needs to decide, e.g., if a contribution is a class or an instance or if a name of an item in the ontology is correct or not. Later, the community uses the classified contributions for query processing.

The general scenario is the same as in the previous chapter. I.e., members of the community create contributions collaboratively and subsequently rate each others' contributions. Ratings correspond to the possible classes of an entry, e.g., "class/instance" or "correct/incorrect" in the ontology example above. After the ratings have been submitted, the community classifies the contributions by aggregating the ratings of the contributions. For the work at hand, we mainly focus on a binary classification setting, i.e., a given contribution can belong to two possible classes.

Note that the open-community setting that we study differs from the typical paid crowdsourcing settings such as Amazon Mechanical Turk (AMT). AMT offers mechanisms to qualify workers by asking them to rate specific contributions. Based on this qualification, workers can be excluded from participating in certain tasks. In contrast to this, we assume a high degree of autonomy of the individual community members. We can neither manipulate individual raters to rate selected contributions nor exclude them from rating.

A simple scheme for aggregating ratings is the well-known majority vote (MV). Despite its simplicity, MV can achieve a surprisingly high accuracy provided that the quality of ratings is sufficiently high [Con85]. One aim of this chapter is to give an intuition for the high performance of MV and to show under which conditions it is achieved. Aggregating ratings by means of the weighted majority vote (WMV) can increase the classification accuracy compared to MV, provided WMV knows the individual competence of each rater, i.e., his[1] probability of rating correctly. Intuitively, weighted majority vote assigns a higher weight to high-competence raters than to low-competence raters. Yet, in general, rater competencies are unknown.

For this case, Dawid and Skene proposed an algorithm (Dawid-Skene algorithm,

[1] To balance things out, we refer to raters in this chapter as male.

DSA) to estimate the competencies of the raters and to classify the contributions accordingly [DS79]. DSA was originally developed to combine opinions of multiple physicians for medical diagnosis. In recent years, there have been a lot of proposals to use DSA – and algorithms closely related to DSA – in particular for crowdsourcing settings [WRW+09, WIP11, RY12, WIP]. However, despite its popularity, DSA has two major shortcomings: (1) it is vulnerable to low-competence settings, and (2) it is defenseless against collusion attacks. In this chapter, we address these two shortcomings.

Firstly, if the mean competence of the community of raters is close to or less than random, e.g., close to or less than 0.5 in binary classification tasks, DSA performs rather poorly. Such a low mean competence can occur if the topic of the community is inherently difficult. It can also occur if the community has a large fraction of spammers, biased raters, malicious raters, or simply raters with consistent misunderstanding. A spammer assigns ratings randomly, independent of the true value of the object rated. Biased raters give consistently too high or too low ratings. [IPW10] gives the following example of a biased rater: Think of a task of classifying web content into the categories appropriate and non-appropriate for children. For this task, parents of young children tend to rate more conservatively than the general population. That is, they tend to consistently rate sites as inappropriate for children that the general population rates as appropriate. Another example might be a community that builds a knowledge base on the advantages and disadvantages of various kinds of power plants. Here, environmental activists would likely be biased in their assessments. For example, they might assess nuclear power plants as more dangerous than they are objectively. Further, raters might give anti-correlated ratings. That is, on average, they invert ratings, either maliciously or because of a consistent misunderstanding. Settings with mean competence close to or less than random seem rare but do occur. For example, Kazai et al. report an average mean competence close to random (0.56) over eight binary open crowdsourcing tasks [KKMF13]. In one of their tasks, the workers reached only a mean competence of 0.35. Such low-competence settings can have a devastating effect on the classification accuracy of DSA and related approaches which assume that the majority is correct on average.

Secondly, DSA is defenseless against collusion attacks. In a collusion attack, raters coordinate to rate the same data objects with the same value to artificially increase their estimated competence. This is beneficial for the colluders if they receive a remuneration for their ratings that is based on their estimated competence. For example, many online communities remunerate users with reputation or karma points for high-quality contributions. We propose to compute remunerations in such communities contingent on the competence estimates by DSA. Similarly, Wang et al. propose an algorithm that pays crowdsourcing workers based on, among other metrics, the competence estimates calculated by DSA [WIP]. However, they do not address collusion attacks. When the remuneration is based on the estimated competence, a collusion attack allows colluders to artificially increase their remuneration while saving cognitive effort for determining the truthful value of the data objects. Since DSA assigns an inflated weight to their low-competence ratings, colluders can also severely damage the accuracy of DSA.

We propose gold strategies based on the level of agreement to increase the accuracy of DSA in low-competence settings and to counter collusion attacks. Gold strategies

adopt the notion of gold objects, i.e., contributions that DSA knows the true value of. Gold objects are a common approach in the literature to differentiate between high- and low-quality raters. The approaches known in the literature use predetermined, randomly selected gold objects. However, as we will show, simply selecting contributions as gold objects at random does not increase the accuracy to a satisfying degree in our setting. Instead, our gold strategies select contributions based on the ratings they have received. Specifically, our gold strategies select contributions based on the level of agreement between community members, i.e., to what extent their ratings agree on the class of a given contribution. Subsequently, trusted experts rate the selected contributions, thereby turning them into gold objects. Of course, the accuracy benefit of gold objects is offset by their costs. Consequently, we are interested in maximizing the net benefit of gold objects, i.e., the benefit of a given number of gold objects minus their costs. Determining the number of gold objects that maximizes the net benefit a priori is infeasible. We propose an algorithm that adaptively determines the number of gold objects based on runtime information.

To summarize, in this chapter we investigate the following:

- *Properties of MV and WMV.* We discuss the relationship between the number of raters, the competence distribution, and the accuracy for MV and WMV under different assumptions w.r.t. the competence of raters.

- *Estimation quality of DSA.* We study the effect of the competence distribution and the number of ratings on the estimation quality of DSA in different settings.

- *Gold strategies based on the level of agreement.* We propose gold strategies, i.e., selecting contributions for evaluation by expert raters, that use the level of agreement between the ratings of the community as a selection criterion. We test the effectiveness of the gold strategies in various settings by means of simulation.

- *Adaptive Gold Algorithm.* We propose an algorithm that determines the number of gold objects in order to maximize the net benefit. Instead of fixing a predetermined number of gold objects, the adaptive algorithm automatically decides when to stop adding further gold objects based on runtime information. We evaluate the algorithm by means of simulation.

- *Collusion attacks.* We study the effects of collusion attacks against DSA. Further, we test the effectiveness of gold strategies to reduce the benefit gained by colluding. To the best of our knowledge, we are the first to study collusion attacks against DSA.

A main finding of ours is that gold strategies based on a high level of agreement between raters improve the accuracy of DSA in low-competence settings considerably. Moreover, the adaptive gold algorithm reaches over 90 percent of the net benefit that the respective gold strategy can maximally achieve. Finally, we find that gold strategies are highly effective in countering collusion attacks against DSA.

Our analysis concentrates on DSA. However, we argue that the related methods that are based on DSA potentially suffer in low-competence settings and under collusion attacks as well. This is because these methods either (1) use the output of DSA as input for their algorithm [IPW10], or (2) they make assumptions similar to those of DSA (even though with some modifications); like DSA, they use an expectation-maximization framework to estimate competencies and to classify the rated items [WRW+09, RY12]. Thus, our findings are somewhat orthogonal to DSA and applicable in principle to these related methods as well.

This chapter is structured as follows. Section 4.1 introduces the formal model and the notation. Section 4.2 discusses the accuracy of majority voting and related decision rules. Section 4.3 presents the DSA algorithm. Section 4.4 introduces two basic simulation settings we use as templates for the evaluation of DSA. Section 4.5 analyzes the effects of the number of data objects and the mean competence of the raters on the accuracy of DSA. Section 4.6 discusses the gold strategies and evaluates them. Section 4.7 introduces and evaluates the adaptive gold algorithm. Section 4.8 discusses the effects of collusion attacks against DSA and evaluates the gold strategies to counter them. Section 4.9 reviews related work. Section 4.10 discusses the main findings and Section 4.11 concludes.

4.1. Model and Notation

We consider an online community where participants can rate the contributions of their peers. Let $K = \{1, \ldots, m\}$ denote the set of contributions and let $k \in K$ denote a single contribution. We also call a contribution a data object in the following. We assume that each data object has a fixed type t from the set of types T.[2] We focus on a binary setting. I.e., there are two different types (for example 'correct/incorrect', 'class/instance' etc.). We encode the types with -1 and 1, i.e., $T = \{-1, 1\}$.

We use o_k to denote the true type of data object k and $o_k = t$ to denote the event that the true type of k is t. Let $p(t)$ denote the prior probability of a randomly chosen data object to be of type t. Raters are those participants of the online community who issue ratings. We use $I = \{1, \ldots, n\}$ to denote the set of all n raters of the community.

Let $r_{i,k} \in T$ denote the rating given by rater i to data object k. We use $R = \{r_{i,k}\}$ to denote the set of all ratings and $s = |R|$ to denote the number of all ratings. Each rater rates each data object at most once. We use $R_k = \{r_{i,k'} \mid k' = k\}$ to denote the set of ratings for data object k and $s_k = |R_k|$ to denote the number of ratings for k.

A classification method estimates the type $\hat{o}_k \in T$ of each data object k. We use $\hat{o}_k = o_k$ and $\hat{o}_k \neq o_k$ to denote the events that the classification method estimates the type of o_k correctly and incorrectly, respectively. Throughout this chapter, we use $\hat{\theta}$ to indicate an estimator of a parameter θ. Finally, let $\mathbb{1}(\cdot)$ be the indicator function, i.e., $\mathbb{1}(\cdot)$ is equal to one if its argument holds true, and equal to zero otherwise. See Table C.1 in Appendix C for a summary of the notation.

[2] In line with the HRMs described in Section 2.2, we use the term 'type' instead of 'class' here.

4.1.1. Competence Models

A rater perceives the type of a data object with some error and rates it according to his perception. Note that in contrast to HRMs, we assume here that raters do not act strategically. I.e., we do not consider the case where raters might change their behavior depending on the behavior of other raters. (We do consider this case below in Section 4.8 when discussing collusion attacks.) Let $P(r_{i,k} = q \mid o_k = t)$ denote the *response probability* that rater i gives a rating of value $q \in T$ given that the true type of the rated data object is $t \in T$. We assume that $P(r_{i,k} = q \mid o_k = t)$ is the same for all data objects k of type t. In other words, for a given rater all data objects of a given type are equally difficult. Further, we assume that, conditional on the type of a data object, ratings are independent and identically distributed.

We use the following three models to capture assumptions of increasing strictness on the response probability.

Type-Dependent Competence. We call the probability that rater i rates correctly given that the true type of the data object is t, i.e., $c_i^{(t)} = P(r_{i,k} = t \mid o_k = t)$, i's *competence* for type t. Since we consider binary types, i.e., $t \in \{-1, 1\}$, it follows that $P(r_{i,k} = t \mid o_k = t) = 1 - P(r_{i,k} = q \mid o_k = t)$ for $t \neq q$. Thus, the set of competencies $\{c_i^{(-1)}, c_i^{(1)}\}$ specifies all response probabilities of rater i.

Heterogeneous Type-Independent Competence. We assume that rater i rates correctly with competence $c_i = c_i^{(t)}$ that is the same for both types $t \in \{-1, 1\}$.

Homogeneous Competence. We assume that every rater i has the same type-independent competence $c = c_i$.

Having defined the competence, we can clarify the notion of spammers, biased, and anti-correlated raters introduced above. A biased rater's competence is low for one type only. An anti-correlated rater inverts ratings, either because he is malicious or he consistently mixes up both categories. This means that his competence for both types is less than 0.5. A spammer has competence 0.5.

4.2. The Accuracy of Majority Decision Rules

To gain insights into the relationship between rater competence and classification accuracy we start by investigating the accuracy of the following majority decision rules: majority vote, and maximum a posteriori probability (MAP) rule, as well as the weighted majority vote as a special case of the MAP rule.

4.2.1. Majority Vote

MV decides for type t if more than one half of the ratings are in favor for t [3]

$$\hat{o}_k = t \text{ if } \sum_{r_{i,k} \in R_k} \mathbb{1}(r_{i,k} = t) \geq \left\lfloor \frac{s_k}{2} \right\rfloor + 1 \tag{4.1}$$

where $\lfloor s_k/2 \rfloor$ denotes the 'floor' under $s_k/2$, i.e., the largest integer smaller than $s_k/2$.

4.2.2. Accuracy of Majority Vote for Homogeneous Competencies

First, we discuss the case where all raters have the same competence c, i.e., $c = c_i$ for all i. Later, we will drop this assumption.

MV classifies a contribution correctly if more than half of the ratings are correct (cf. Equation 4.1). For example, suppose that we want to classify a contribution based on three ratings. In this case, MV decides correctly if exactly two ratings are correct, for which there are three possible ways, or if exactly three ratings are correct. Thus, the probability of correct classification is the sum of two terms: $3c^2(1-c) + c^3$. The general formula for the accuracy of MV under homogeneous competence can be derived by summing up the probabilities that l_k out of s_k ratings are correct for all $l_k \geq \lfloor s_k/2 \rfloor + 1$, i.e.,

$$P_{\text{MV}}(\hat{o}_k = o_k) = \sum_{l_k = \lfloor s_k/2 \rfloor + 1}^{s_k} \binom{s_k}{l_k} (c)^{l_k} (1-c)^{s_k - l_k}. \tag{4.2}$$

Figure 4.1 illustrates the relationship between the accuracy of MV and the competence for different odd numbers of raters.

The relationship between the competence, the number of raters, and the accuracy of MV has first been formulated by Marquis de Condorcet in 1785 [Con85] and is known as Condorcet's jury theorem (CJT): For odd s_k and homogeneous rater competence c

- if $c > 0.5$, then $P_{\text{MV}}(\hat{o}_k = o_k)$ increases monotonically in s_k, and $\lim_{s_k \to \infty} P_{\text{MV}}(\hat{o}_k = o_k) = 1$,
- if $c < 0.5$, then $P_{\text{MV}}(\hat{o}_k = o_k)$ decreases monotonically in s_k, and $\lim_{s_k \to \infty} P_{\text{MV}}(\hat{o}_k = o_k) = 0$,
- if $c = 0.5$, then $P_{\text{MV}}(\hat{o}_k = o_k) = 0.5$ for all s_k.

Grofman et al. [GOF83] show that the CJT is also valid for heterogeneous type-independent competencies c_i if the distribution of c_i is symmetric around the mean $\bar{c}$. In that case, $\bar{c}$ substitutes c in the CJT.

[3] The rule that decides in favor for the type t that receives most of the ratings, i.e., $\hat{o}_k = t$, if $\arg\max_{t \in T} = \sum_{r_{i,k}} \mathbb{1}(r_{i,k} = t)$, is called plurality vote. In the literature simple majority vote and plurality vote are often both called majority vote according to [Kun04]. Plurality vote and majority vote are equivalent for settings where the number of types is two and the number of ratings is odd.

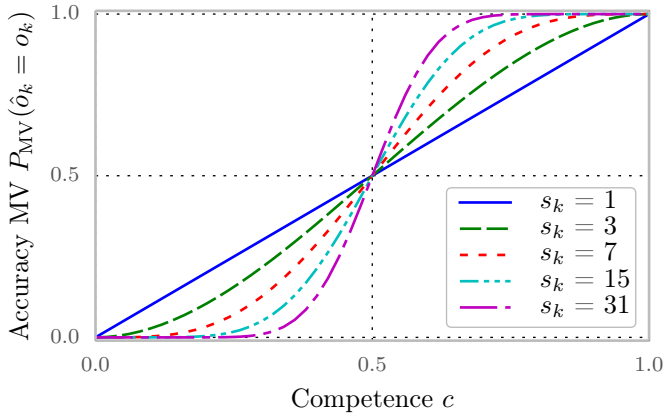


Figure 4.1.: Accuracy of majority vote for homogeneous competence and odd numbers of ratings s_k.

Optimality of Majority Vote for Homogeneous Competencies greater than 0.5

Even though it is a simple method, MV achieves a high accuracy, provided that the competence is greater than 0.5. In fact, if we add the assumption that the prior is not too heavily skewed in favor of one type, we can show that MV is the optimal classification scheme for two-type settings.

Proposition 1 (Optimality of Majority Vote under Homogeneous Competence)**.** *Under the assumption of the homogeneous competence model in Section 4.1.1 and given that s_k is odd, $c > 0.5$, and $1 - c < p(t) < c$ for both types $t \in \{-1, 1\}$, majority vote is an optimal decision rule, i.e., there exists no decision rule that has a higher accuracy.*

The proof is in Appendix B.1.

4.2.3. MAP Rule and Weighted Majority Vote for Known Competencies and Type Priors

The MAP rule and WMV are restatements of Bayes' theorem. Thus, both methods yield optimal estimates of the data object types $\hat{o}_k$, assuming that they know the true competencies and the true type prior. This is rarely satisfied in the real world. However, discussing these methods reveals insights into the behavior of DSA.

Maximum a Posteriori Probability Rule

The MAP rule considers the type dependent competence model with known competencies. It decides in favor of the type t with the maximum posterior probability given the ratings, i.e.,

$$\hat{o}_k = \arg\max_{t \in T} P(o_k = t \mid R_k). \tag{4.3}$$

Ties are handled arbitrarily. Since $T = \{-1, 1\}$, this is equivalent to deciding in favor of the type with the higher posterior log-odds

$$\hat{o}_k = \text{sign}\left(\log \frac{P(o_k = 1 \mid R_k)}{P(o_k = -1 \mid R_k)}\right).$$

By Bayes' theorem, the posterior probability that data object k is of type t given the ratings R_k is

$$P(o_k = t \mid R_k) = \frac{P(R_k \mid o_k = t) \cdot p(t)}{P(R_k)}.$$

Since we assume conditional independence of the ratings given the type of the data object, the likelihood of the ratings can be expressed as the product of the individual likelihoods $P(R_k \mid o_k = t) = \prod_{r_{i,k} \in R_k} P(r_{i,k} \mid o_k = t)$. Thus, the posterior log-odds for data object k are

$$\log \frac{P(o_k = 1 \mid R_k)}{P(o_k = -1 \mid R_k)} = \log \frac{p(1)}{p(-1)} + \sum_{r_{i,k} \in R_k} \log \frac{P(r_{i,k} \mid o_k = 1)}{P(r_{i,k} \mid o_k = -1)}. \tag{4.4}$$

In other words, in our two-type model, the MAP rule is equivalent to a majority vote that weighs each rating by the log ratio of the rating likelihoods of its rater:

$$\hat{o}_k = \text{sign}\left(\log \frac{p(1)}{p(-1)} + \sum_{r_{i,k} \in R_k} \log \frac{P(r_{i,k} \mid o_k = 1)}{P(r_{i,k} \mid o_k = -1)}\right). \tag{4.5}$$

That is, a rating $r_{i,k} = q$ has weight $\log(P(r_{i,k} = q \mid o_k = 1)/P(r_{i,k} = q \mid o_k = -1))$. Later, we will use this insight to derive a weighted measure of the level of agreement between raters.

Weighted Majority Vote with Optimal Weights

WMV with optimal weights is a special case of the MAP rule. It assumes known type-independent heterogeneous competence $c_i = c_i^{(t)} = P(r_{i,k} = t \mid o_k = t)$ for all types t and all data objects k. Thus, we can reformulate the individual likelihoods

$$\frac{P(r_{i,k} \mid o_k = 1)}{P(r_{i,k} \mid o_k = -1)} = \begin{cases} c_i/(1 - c_i) & \text{if } r_{i,k} = 1 \\ (1 - c_i)/c_i & \text{if } r_{i,k} = -1. \end{cases}$$

Since $\log c_i/(1 - c_i) = -\log(1 - c_i)/c_i$ we can restate Equation (4.5) as the WMV rule

$$\hat{o}_k = \text{sign}\left(\log \frac{p(1)}{p(-1)} + \sum_{r_{i,k} \in R_k} r_{i,k} v_i(c_i)\right)$$

with

$$v_i(c_i) = \log \frac{c_i}{(1 - c_i)}. \tag{4.6}$$

being the optimal weight of rater i.

Equation (4.6) reveals a relationship between the (type-independent) competence of a given rater i and i's usefulness for the estimation of the data object type. Raters with competence $c_i > 0.5$ have positive weight. Raters with competence $c_i < 0.5$ give on average the opposite of the true rating, either maliciously or because of consistent misunderstanding. The function v_i reverses the "direction" of their ratings by assigning them a negative weight. Moreover, $v_i(c_i) = -v_i(1-c_i)$. In other words, a low competence rater i' with $c_{i'} < 0.5$ is equally beneficial for the accuracy of WMV as a high competence rater i'' with $c_{i''} = 1 - c_{i'}$. Spammers, i.e., raters with competence near 0.5, on the other hand, are worst for the accuracy of WMV. This is because they generate ratings that are completely random. They have zero weight, i.e., $v_i(0.5) = 0$.

4.2.4. Accuracy of Majority Vote vs. Accuracy of Weighted Majority Vote with Known Rater Competencies

We explore the influence of the number of raters and of the competence distribution on the accuracy of WMV. To this end, we run a simulation under the type-independent heterogeneous competence model. We draw the competence c_i of each rater i uniformly at random from the interval $[\bar{c} - w/2, \bar{c} + w/2]$, with mean competence $\bar{c}$ and interval width w. For example, for $w = 0.4$, and $\bar{c} = 0.5$ we draw competencies uniformly from the interval $[0.3, 0.7]$. We vary the mean competence $\bar{c}$ from $0 + w/2$ to $1 - w/2$ in 0.05 steps. We use a uniform prior, i.e., $p(-1) = p(1) = 0.5$ and average the results over 100 simulation runs. The simulation breaks ties by fair coin toss. In the simulation we use the competence c_i to compute i's optimal weight. Since the prior $p(t)$ is uniform, it has no influence on the type estimates. Figure 4.2 shows the accuracy of MV and WMV as a function of the mean competence $\bar{c}$ for different numbers of raters n and different interval widths w. For $\bar{c} > 0.5$ the accuracy of WMV is higher than or as high as the accuracy of MV. Like MV, WMV benefits from higher n.

The accuracy curves are minimal at $\bar{c} = 0.5$ and symmetric w.r.t. the line $\bar{c} = 0.5$. In other words, the further $\bar{c}$ is away from 0.5, the higher the accuracy. This is because high-competence and low-competence raters are equally beneficial for the accuracy of WMV, as described above. Competencies near 0.5, on the other hand, are worst for the accuracy of WMV because they generate random ratings. This also explains why WMV benefits from larger competency ranges w. Namely, the probability of having raters with very high or with very low competence increases with w.

As already mentioned, WMV with known competencies is optimal (under the model in Section 4.1) because WMV is a restatement of Bayes' theorem. Therefore, it represents an upper bound for the mean accuracy of rating aggregation methods discussed in this chapter, in particular for DSA.

However, WMV assumes that the competencies c_i of raters and the $p(t)$ are known quantities. In open settings, this assumption does not hold. The remainder of this chapter deals with the problem of what to do if competencies are unknown.

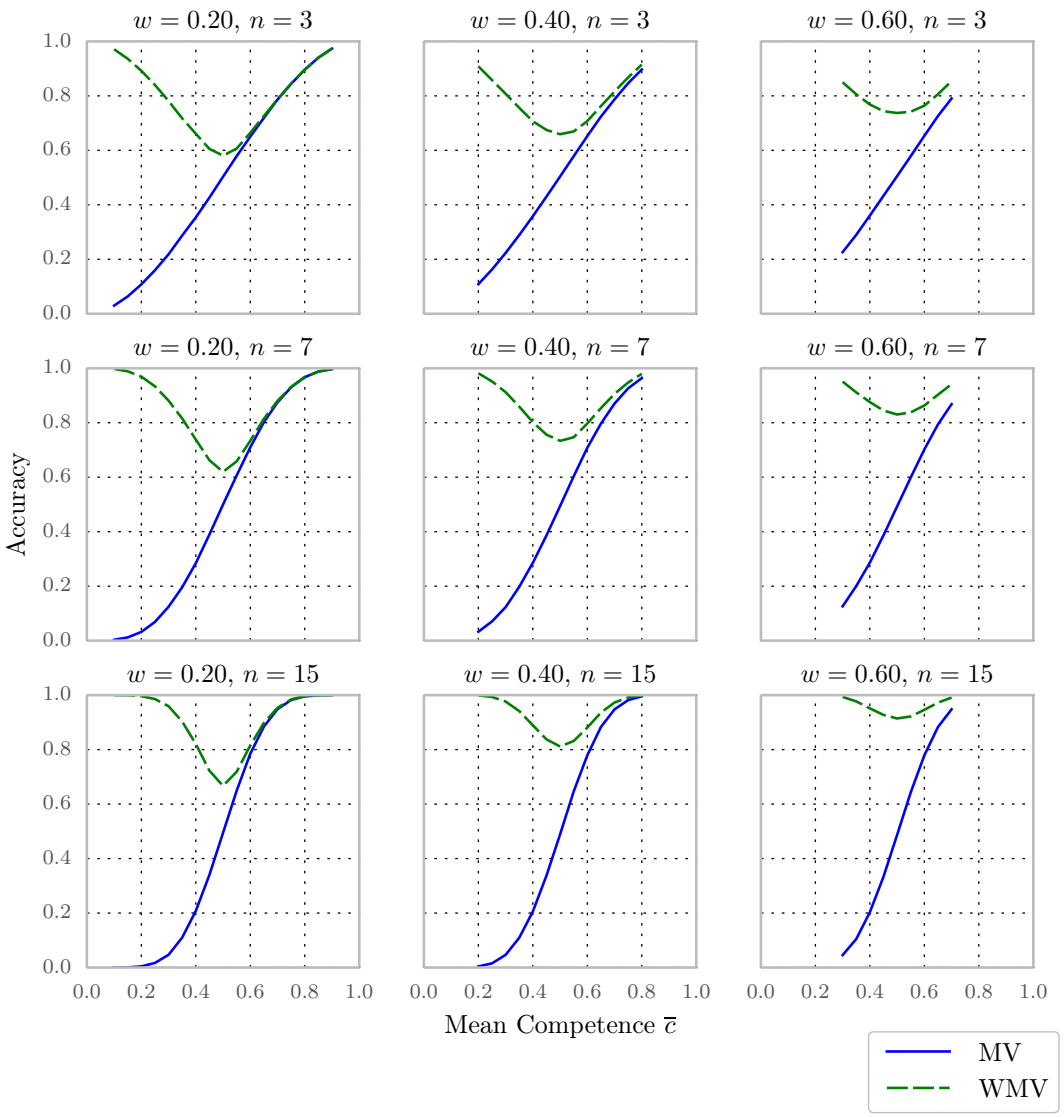


Figure 4.2.: Accuracy of MV and WMV for known competencies c_i for different competence interval widths w and numbers of raters n.

4.3. Estimation of Rater Competencies and Data Object Types with the Dawid-Skene Algorithm

DSA relies on the type dependent competence model.

Using only the ratings as inputs, DSA estimates (1) the competencies $\hat{c}_i^{(t)}$, (2) the type priors $\hat{p}(t)$, and (3) the type probabilities $\hat{P}(o_k = t)$. DSA iterates between computing the competencies (1) and the type priors (2) using the type probabilities (3) as input, and computing the type probabilities (3) using the competencies (1) and the type priors (2) as input.

Algorithm 1: Our implementation of DSA.

Input: set of ratings $\{r_{i,k}\}$ given by rater i to data object k, additive smoothing parameters a and d

Output: set of estimated types $\{\hat{o}_k\}_{k \in K}$, set of estimated response probabilities $\{\hat{P}(r_{i,k} = q \mid o_k = t)\}_{i \in I, q \in T, t \in T}$ (equivalent to set of estimated competencies $\{\hat{c}_i^{(t)}\}_{i \in I, t \in T}$)

1 **foreach** *contribution* $k \in K$ **do**
2 initialize $\hat{P}(o_k = t)$ with majority vote.
3 **repeat**
4 **foreach** *type* $t \in T$ **do**
5 $\hat{p}(t) \leftarrow \dfrac{a + \sum_{k=1}^{m} \hat{P}(o_k = t)}{ad + m}$
6 **foreach** $i \in I$, $q \in T$, $t \in T$ **do**
7 $\hat{P}(r_{i,k} = q \mid o_k = t) \leftarrow \dfrac{a + \sum_{k \in K} r_{i,k}^{(q)} \cdot \hat{P}(o_k = t)}{ad + \sum_{q \in T} \sum_{k \in K} r_{i,k}^{(q)} \cdot \hat{P}(o_k = t)}$
8 **foreach** *contribution* $k \in K$ *and each type* $t \in T$ **do**
9 $\hat{P}(o_k = t) \leftarrow \dfrac{\hat{p}(t) \prod_{i \in I} \prod_{q \in T} \hat{P}(r_{i,k} = q \mid o_k = t)^{\mathbb{1}(r_{i,k} = q)}}{\sum_{t \in T} \hat{p}(t) \prod_{i \in I} \prod_{q \in T} \hat{P}(r_{i,k} = q \mid o_k = t)^{\mathbb{1}(r_{i,k} = q)}}$
10 **until** $\{\hat{P}(r_{i,k} = q \mid o_k = t)\}_{i \in I, q \in T, t \in T}$ *converges*;
11 **foreach** $k \in K$ **do**
12 $\hat{o}_k \leftarrow \arg\max_{t \in T} \hat{P}(o_k = t)$

For brevity and clarity we use the estimates of the response probabilities $\hat{P}(r_{i,k} = q \mid o_k = t)$ instead of the competence estimates in the following description. Since we use binary types this is equivalent to using the type dependent competencies (cf. Section 4.1.1). Having said this, DSA proceeds as follows (cf. Algorithm 1). It initializes the type estimates for each data object. Then it repeats the following three steps until convergence.

1. It estimates the prior of type t by summing up the estimated probabilities of each data object being of type t and dividing the sum by the number of data objects m

(line 5 in Algorithm 1).

2. To infer the response probability $\hat{P}(r_{i,k} = q \mid o_k = t)$, DSA sums up the ratings i has given in favor for type q and weighs each rating by the estimated probability that the rated data object is of type t. It normalizes the obtained sum by the weighted sum of all ratings of i (line 7).

3. DSA computes the posterior probability that data object k has type t given the ratings it received (line 9). Since it does not know the true response probabilities and the true type prior neccessary for the computation, DSA uses the estimates of these quantities obtained in the two previous steps.

Finally, DSA classifies each data object by assigning the type t that has the maximum estimated posterior probability.

We have added two implementation details that the authors of DSA did not specify. First, the authors leave the initialization of $\hat{P}(o_k = q)$ unspecified. We use MV to this end. Further, we use additive smoothing (line 5 and line 7 of Algorithm 1) to avoid 0 probabilities that would cancel out all other factors of the product in line 9 of Algorithm 1. For the two-type setting in this chapter, we set the smoothing parameters $a = 0.1$ and $d = 2$.

Obtaining the competence estimates $\hat{c}_i$ of the type-independent competence model is straightforward. We simply sum up DSA's estimates of the type dependent competencies and weigh them with the estimates of the type priors

$$\hat{c}_i = \hat{c}_i^{(-1)}\hat{p}(-1) + \hat{c}_i^{(1)}\hat{p}(1). \tag{4.7}$$

In the following we investigate the behavior of DSA.

4.4. Settings of a Simulation to Analyze the Dawid-Skene Algorithm

To gain insights into the behavior of DSA we analyze its performance by means of simulation. This simulation is rather unrelated to the previous one in Section 4.2.4. We use the type-independent heterogeneous competence model. For each rater i we generate random ratings according to his competence c_i. We draw the competence c_i of each rater i uniformly at random from the interval $[\bar{c} - w/2, \bar{c} + w/2]$, with mean competence $\bar{c}$ and interval width w. To describe the number of ratings per rater we introduce a simulation parameter *rating rate*. The rating rate of rater i, $rr_i \in [0, 1]$, is the probability that i assigns a rating to a given data object.

We use two basic simulation settings – UNIFORM and SKEWED – to cover two common scenarios. They differ with respect to the distribution of user ratings, the number of raters, the number of data objects, and the type prior. We have determined the default number of data objects m for each setting by simulating each setting with successively increasing m while keeping the other parameters fixed (see Section 4.5.1). The resulting default m for UNIFORM and SKEWED are the points where the accuracy of DSA starts to converge to a steady state.

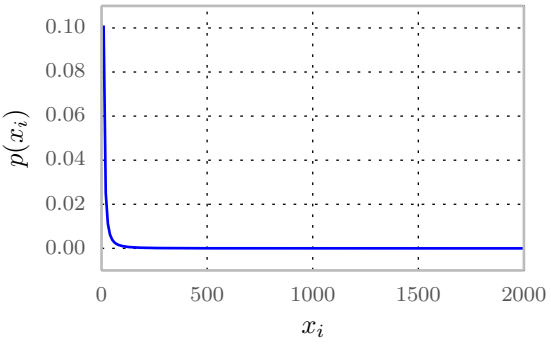


Figure 4.3.: The probability density function of the truncated Pareto distribution of setting SKEWED.

4.4.1. Simulation Setting UNIFORM

This setting represents a small, homogeneous community for example in a company or a lecture community. The default number of raters for this setting is $n = 50$. We also use smaller n in experiments where we want to analyze the effect of n on the accuracy of DSA. The rating rate in this setting is the same for all raters, i.e., $rr = rr_i$ for all raters $i \in I$. We set $rr = 0.4$, that is, on average 2 out of every 5 raters issue a rating for a given data object. The prior for a data object to be of a given type is uniform, i.e., $p(0) = p(1) = 0.5$. We set the default number of data objects to $m = 400$. We average results of this settings over 100 simulation runs with different random seeds.

4.4.2. Simulation Setting SKEWED

This setting represents an open online community with a highly skewed rating rate and a skewed prior. To this end we draw the number of ratings per rater from a Power-law distribution. The main characteristic of such a distribution is that most ratings come from a small fraction of the raters while most raters issue only very few ratings each. Power-law distributions are frequently observed in open online communities [MMM+11, MJD09]. See also Section 2.1.1. Since the number of ratings is bounded by the number of data objects, we draw the number of ratings x_i for each rater i from a truncated Pareto distribution [AMP06] defined by the density function

$$p(x_i) = \frac{\alpha x_{\min}^{\alpha} x_i^{-\alpha-1}}{1 - (x_{\min}/x_{\max})^{\alpha}}$$

for $0 < x_{\min} \leq x_i \leq x_{\max} < \infty$, where $x_{\min} < x_{\max}$. The upper bound is $x_{\max} = m$ since each rater can issue at most one rating per data object. We set m to 2000 for this setting. Further, we set the lower bound for the number of ratings per rater to $x_{\min} = 10$. Estimating the competencies of raters who have issued less than 10 ratings becomes unnecessarily inaccurate (we argue). We set the shape parameter of the distribution to a typical value of $\alpha = 1$. To obtain discrete values for the number of ratings we round to the nearest integer [CSN09]. This results in a highly right-skewed distribution of x_i

with a skewness of approx. 5.29 and a mean rating rate of approx. 0.026 (see Figure 4.3). We set the prior for this setting to $p(1) = 0.7$ because we assume that a typical online community has an uneven ratio of good vs. bad data objects. Finally, we set the number of raters to $n = 100$.[4] As in UNIFORM we average the results of 100 simulation runs with different random seeds.

4.5. Analyzing the Estimation Quality of the Dawid-Skene Algorithm

We conduct a series of simulation experiments to find out how accurately DSA estimates

- the true type of the data objects, and
- the competence of the raters.

In the following, we call the ratio of data objects that DSA classifies correctly *classification accuracy* or simply *accuracy*. Alternatively, we measure the error rate, which equals one minus the accuracy. Further, we measure the error rate of the competence estimates. To this end, we define the mean absolute difference between the competencies of the raters and their estimated competencies

$$madc = \sum_{i=1}^{n} |c_i - \hat{c}_i| / n.$$

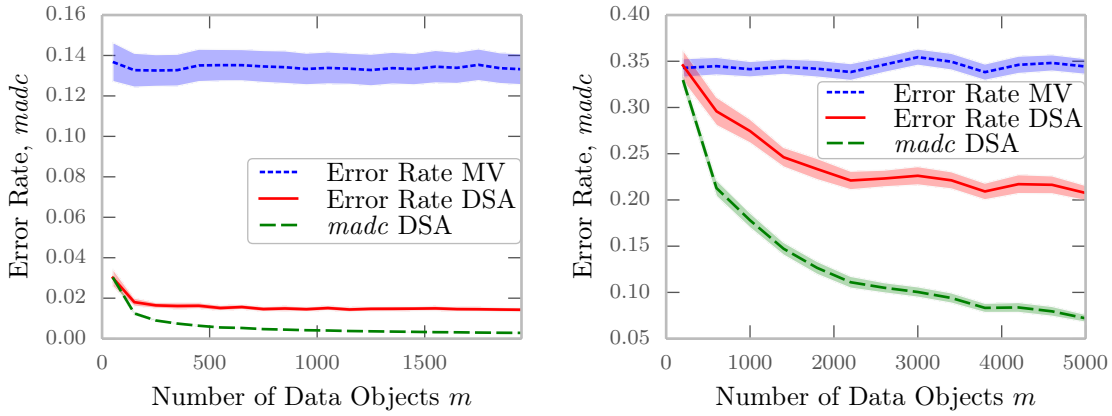
where $\hat{c}_i$ denotes the estimate of the type-independent competence of rater i obtained by means of Equation (4.7).

4.5.1. Effect of the Number of Data Objects on the Estimation Quality of the Dawid-Skene Algorithm

How does the number of data objects m influence the estimation quality of DSA? Or, put differently: If the community size stays constant but the number of contributions created and rated by the community grows over time, then how many contributions does it take for DSA's accuracy and *madc* to stabilize? To answer this question, we conduct one simulation experiment for each basic setting. For both settings we draw the c_i uniformly at random from the interval $[0.3, 0.95]$.

Figure 4.4 shows the effect of the number of ratings on the error rate and *madc*. For setting UNIFORM (see Figure 4.4a) the error rate as well as *madc* of DSA converge for $m > 400$. For SKEWED we set the lower bound for the number of ratings per rater proportional to m, $x_{min} = m/200$, to keep the rating rate approximately equal for different values of m. The DSA error rate in SKEWED shows less convergent behavior (see Figure 4.4b). Nevertheless, there is a strong reduction of the error rate and the *madc*

[4] We chose $n = 100$, since we deem smaller, more unstable communities the more interesting case. Further, our results (not shown) indicate, that larger n do not change the simulation results to a significant degree.



(a) Setting UNIFORM (b) Setting SKEWED

Figure 4.4.: Error rate and mean absolute difference between estimated and real competence *madc*. Bands show 95% percent confidence intervals.

of DSA for increasing numbers of data objects up to about 2000. Consequently, in the following simulations we set $m = 400$ for UNIFORM and $m = 2000$ for SKEWED.

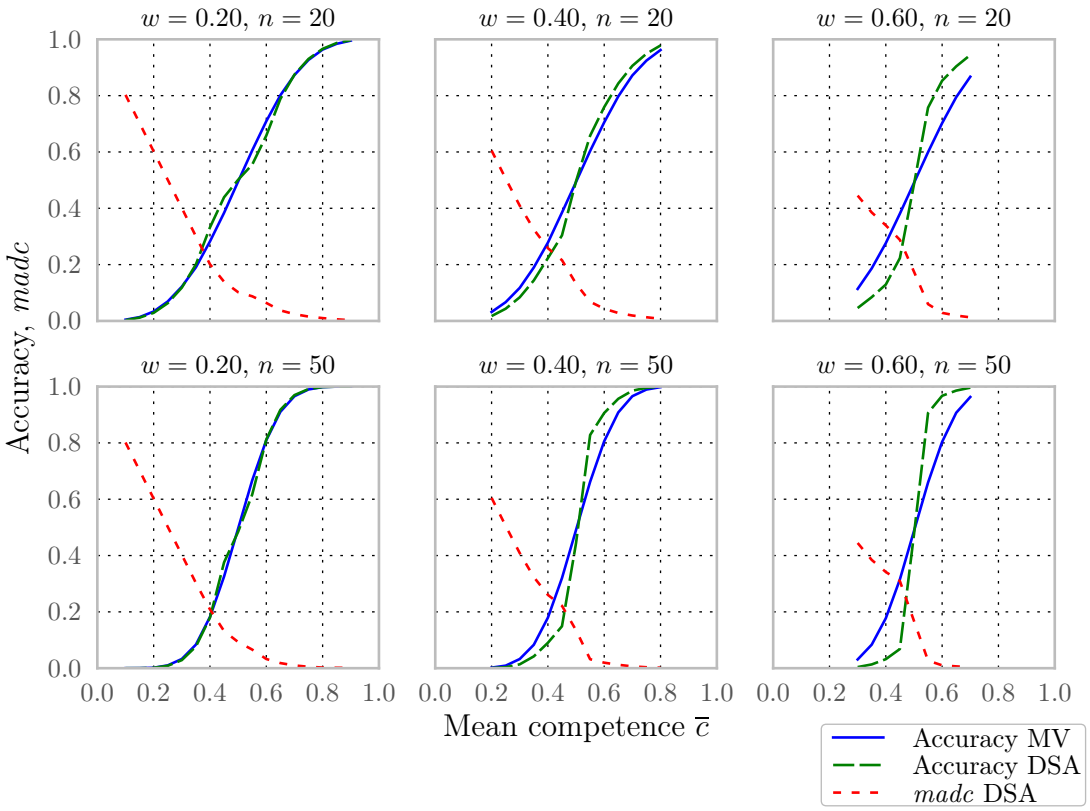
In simple terms, this means that in more homogeneous settings, where raters have roughly the same rating rate and the rating rate is high, DSA stabilizes relatively early. In a more open setting with a low rating rate and a skewed rating distribution that is typical for open internet communities, this is different. Here, DSA requires a much higher number of data objects to stabilize.

4.5.2. Effects of the Number of Raters and of the Competence Distribution on the Estimation Quality of the Dawid-Skene Algorithm

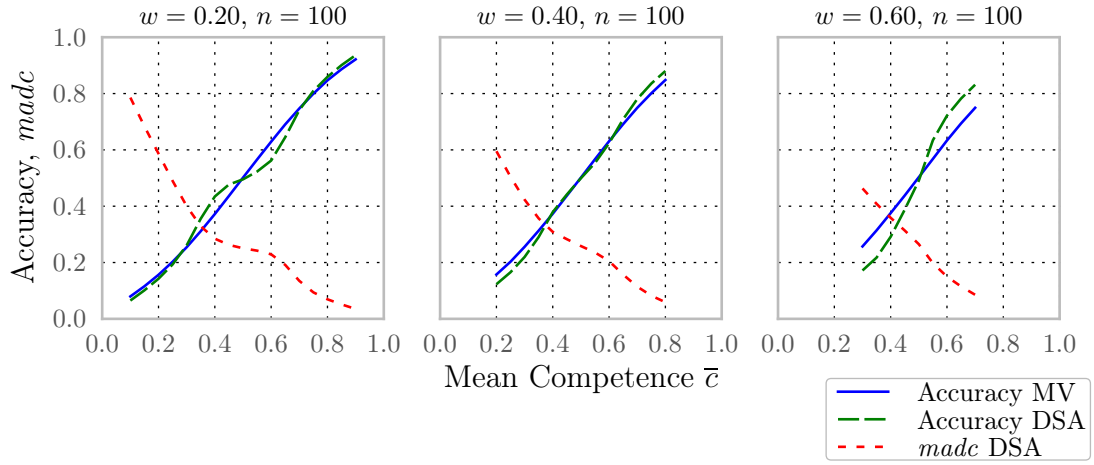
As we have seen, the accuracies of MV and WMV depend on the number of raters and on their competencies (Section 4.2.4). To find out how these two parameters affect the estimation accuracy of DSA, we conduct a simulation experiment using the same procedure as in Section 4.2.4. That is, we draw the competencies of the n raters uniformly at random from the interval $[\bar{c} - w/2, \bar{c} + w/2]$, with mean $\bar{c}$ and interval width w. We vary the mean competence $\bar{c}$ from $0 + w/2$ to $1 - w/2$ in 0.05 steps. The accuracy of MV serves as a baseline.

Unsurprisingly, DSA's accuracy increases with increasing mean competencies (Figure 4.5). For the narrow competence range $w = 0.2$, the accuracies of MV and DSA are similar.

For higher w, DSA's accuracy differs markedly from MV's accuracy. Here, depending on the mean competence $\bar{c}$, DSA's accuracy is either (1) worse than MV's accuracy, for $\bar{c} < 0.5$, or (2) better than MV's accuracy, for $\bar{c} > 0.5$. The reason for this effect is that DSA uses the type estimates $\hat{P}(o_k = t)$ to compute the competence estimates $c_i^{(t)}$, and vice versa. For $\bar{c} < 0.5$, DSA performs worse than MV because in this case the type estimates are highly inaccurate. This in turn causes DSA to invert the competency



(a) Setting UNIFORM



(b) Setting SKEWED

Figure 4.5.: Accuracy and *madc* as a function of the mean competence $\bar{c}$ for different numbers of raters n and different competence interval widths w.

estimates to some degree: it assigns a low competence to high-competence raters and vice versa. This leads to even more inaccurate type estimates, and so on. The opposite is the case for $\bar{c} > 0.5$. Here, DSA estimates the types more accurately. This in turn yields more accurate competence estimates. This is reflected in the error rate of the competence estimates *madc*. For UNIFORM, *madc* drops almost to 0 for $\bar{c} > 0.6$. In that case, DSA's accuracy approaches that of WMV for known rater competencies (Figure 4.2). The higher w and n, the more pronounced this effect becomes.

The curves for the accuracy of both DSA and MV are flatter in the SKEWED setting (Figure 4.5b) than in the UNIFORM setting (Figure 4.5a). This means that, for the same $\bar{c}$, the accuracy in SKEWED is higher than in UNIFORM for $\bar{c} < 0.5$ and lower for $\bar{c} > 0.5$. Consequently, the *madc* curves are flatter in SKEWED than in UNIFORM as well.

4.6. Using Gold Strategies to Increase the Accuracy of the Dawid-Skene Algorithm in Low-Competence Settings

As we have seen, for mean competencies $\bar{c} < 0.5$ DSA's accuracy is low. *Gold objects*, i.e., data objects which we know the true type of, can increase the accuracy of DSA in such low-competence settings. The idea behind using gold objects for DSA is the following. An accuracy of DSA of less than 1.0 means that DSA misclassifies some data objects. Knowing the true type of these data objects with certainty allows DSA to estimate more accurately the competence of the raters who have given ratings to these data objects. This in turn leads to a higher accuracy for the type estimates of non-gold objects these raters have rated. This increases the accuracy of competence estimates even further and so on.

We obtain gold objects by selecting some data objects and letting trusted experts rate these data objects.

We use $K_{\text{gold}} \subseteq K$ to denote the set of gold objects. To integrate gold objects into DSA we use the same straightforward procedure as [WIP11] and simply set the type estimates of the gold objects to their known values at the end of the `repeat until` loop in Algorithm 1. Algorithm 2 shows the resulting modified DSA. For simplicity it shows mostly the modified parts. Dots (...) indicate unchanged parts from Algorithm 1.

Since gold objects are costly we want to use them effectively. For crowdsourcing services such as Amazon Mechanical Turk, Wang et al. propose to achieve this by actively forcing crowdsourcing workers to rate predefined gold objects [WIP11]. However, forcing raters to rate particular contributions is not possible in an open community scenario like ours.

Instead, we propose *gold strategies* to determine which existing contributions of the community to choose as gold objects. We use the following procedure to incorporate gold strategies into an open community scenario. (1) Select contributions as gold objects based on the selection criterion of the respective gold strategy. (2) Determine the true type of these gold objects by means of trusted experts.[5] (3) Run the modified DSA

[5] For domains where we do not trust experts to be completely accurate we could combine the ratings of several experts, for example by means of DSA, to achieve a higher accuracy.

Algorithm 2: Modified Dawid-Skene algorithm with gold objects.

```
/* dots (...) indicate unchanged parts from Algorithm 1.          */
```

Input: ...; set of known types for gold objects $\{o_{k'}\}_{k' \in K_{\text{gold}}}$

```
1  ...
2  repeat
3  |  ...
4  |  foreach contribution k' ∈ K_gold and each type t ∈ T do
5  |  |  if o_k' = t then
6  |  |  |  P̂(o_k' = t) ← 1
7  |  |  else
8  |  |  |  P̂(o_k' = t) ← 0
9  until {P̂(r_i,k = q | o_k = t)}_i∈I,q∈T,t∈T converges;
10 ...
```

(Algorithm 2) with the gold objects to estimate the types of all contributions.

The most straightforward of the gold strategies, the UNI strategy, selects gold objects uniformly at random.

4.6.1. Gold Strategies Based on the Level of Agreement

Additionally to UNI, we propose gold strategies that take the *level of agreement* between raters into account, i.e., to what extent the raters agree on the type of a given data object. The strategies select data objects as gold objects that either have a high (HI) or a low (LO) level of agreement. The rationale for using a high level of agreement is the following. In low-competence communities, i.e., communities with $\bar{c} < 0.5$, a high level of agreement on a type t of data object k indicates that t is likely not the correct type of k. This is because raters with competence less than 0.5 have a higher chance of being incorrect than of being correct. Thus, DSA will likely compute the estimate $\hat{P}(o_k = t)$ inaccurately and, as a consequence, inaccurately estimate the competencies of the raters who have rated k. So the benefit of selecting k as a gold object is potentially high. Conversely, if $\bar{c} > 0.5$, the probability of estimating $\hat{P}(o_k = t)$ inaccurately is highest for data objects with a low level of agreement. Below, we quantify the exact error probability given a certain level of agreement for a simplified setting using MV and homogeneous competence.

In this section, we introduce two methods to measure the level of agreement: the absolute rating sum (ARS) and the estimated absolute posterior log-odds (ALO). Based on the two measures, we define four gold strategies: HI-ARS, LO-ARS, HI-ALO, LO-ALO (prefix HI-/LO- for the level of agreement).

In the following, we use g to denote the gold ratio. The gold ratio is the ratio of gold objects among all data objects, i.e., $g = m_{\text{gold}}/m$, where $m_{\text{gold}} = |K_{\text{gold}}|$ is the number of gold objects.

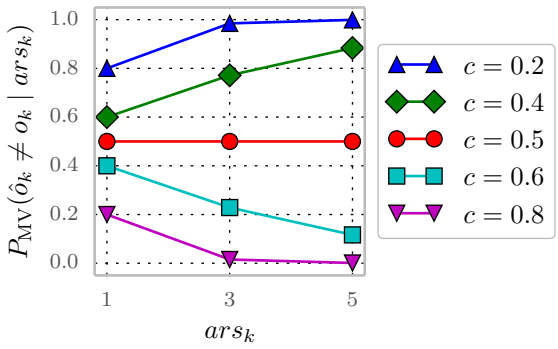


Figure 4.6.: Probability of incorrect classification by MV given the absolute rating sum (ars_k) for five raters with homogeneous competence c.

Gold Strategies Based on the Absolute Rating Sum

We define the absolute rating sum of data object k as $ars_k = \left|\sum_{r_{i,k} \in R_k} r_{i,k}\right|$, where $r_{i,k} \in \{-1, 1\}$. Low and high values of ars_k indicate low and high levels of agreement, respectively. Consequently LO-ARS selects the first m_{gold} data objects ordered by ars_k ascending, while HI-ARS selects the first m_{gold} data objects ordered by ars_k descending.

As an example, consider a setting with two data objects 1 and 2 and their ratings $R_1 = \{1, -1, 1\}$ and $R_2 = \{1, 1, -1, -1\}$. Assume a gold ratio of $g = 0.5$, i.e., half of the data objects are selected as gold objects. The absolute rating sums are $ars_1 = 1$ and $ars_2 = 0$. Thus, HI-ARS selects data object 1 as gold object while LO-ARS selects object 2.

Error probability given ars_k using MV and homogeneous competence. Above we have given an intuition why the level of agreement is a useful criterion for selecting data objects as gold objects: it can identify data objects whose type DSA will likely estimate inaccurately. To further strengthen this intuition, we now quantify – even though only for a simplified setting – the probability that MV will classify a data object incorrectly given the absolute rating sum. We use MV because (1) it allows for an analytical quantification, and (2) it has roughly a similar accuracy as DSA for a given mean competence (see Section 4.5).

Proposition 2. *Let an odd number of ratings s_k for data object k, and homogeneous competence c be given. Then, the probability that MV estimates the type of k incorrectly given the absolute rating sum ars_k is*

$$P_{\mathrm{MV}}(\hat{o}_k \neq o_k \mid ars_k) = \frac{c^{l_k} \cdot (1-c)^{s_k - l_k}}{c^{l_k} \cdot (1-c)^{s_k - l_k} + (1-c)^{l_k} \cdot c^{s_k - l_k}}$$

where $l_k = (s_k - ars_k)/2$ is the number of correct ratings for k.

See Appendix B.2 for the proof of Proposition 2.

Figure 4.6 illustrates this relationship for $s_k = 5$. Depending on the competence c, the probability of incorrect classification either increases or decreases with an increasing

absolute rating sum: It increases for $c < 0.5$, and it decreases for $c > 0.5$. This is what the intuition suggests. If the (mean) competence is below 0.5, a high level of agreement indicates a high error probability. Conversely, if the competence is above 0.5, a low level of agreement indicates a high error probability.

Gold Strategies Based on the Estimated Absolute Posterior Log-Odds

The absolute posterior log-odds for data object k are the absolute value of the posterior log-odds (see Equation (4.4) for the posterior log-odds)

$$\begin{aligned} alo_k &= \left| \log \frac{P(o_k = 1 \mid R_k)}{P(o_k = -1 \mid R_k)} \right| \\ &= \left| \log \frac{p(1)}{p(-1)} + \sum_{r_{i,k} \in R_k} \log \frac{P(r_{i,k} \mid o_k = 1)}{P(r_{i,k} \mid o_k = -1)} \right|. \end{aligned}$$

Like the posterior log-odds, the absolute posterior log-odds are a weighted measure of the agreement level. In addition to summing up the ratings like ars_k, alo_k weighs each rating by the log ratio of the response likelihoods of its rater. See Section 4.2.3 for a discussion of the weights.

We cannot calculate alo_k directly because the true competencies and the true type priors are unknown parameters. Instead we run DSA to obtain estimates of the posterior probability $\hat{P}(o_k = t \mid R_k) = \hat{P}(o_k = t)$ for data object k being of type t (see line 9 of Algorithm 1) and use these estimates to calculate the estimate of alo_k

$$\widehat{alo_k} = \left| \log \frac{\hat{P}(o_k = 1 \mid R_k)}{\hat{P}(o_k = -1 \mid R_k)} \right|.$$

High values of $\widehat{alo_k}$ indicate a high level of agreement for data object k, while low values indicate a low level of agreement. Thus, LO-ALO selects the first m_{gold} data objects ordered by $\widehat{alo_k}$ ascending, and HI-ALO selects the first m_{gold} data objects ordered by $\widehat{alo_k}$ descending.

4.6.2. Evaluation of Gold Strategies

How much does the accuracy of DSA benefit from the different gold strategies? Ideally, the use of gold objects increases the number of non-gold objects that DSA classifies correctly. To study to which extent this indeed occurs, we define the net accuracy as the ratio of correctly classified non-gold data objects

$$netacc = \frac{\sum_{k \in K \setminus K_{\text{gold}}} \mathbb{1}(\hat{o}_k = o_k)}{|K \setminus K_{\text{gold}}|}. \tag{4.8}$$

To quantify the accuracy gains of DSA with gold objects compared to the vanilla DSA without gold objects, we do the following. For each $\bar{c}$ and each gold strategy, we calculate the net accuracy that DSA achieves using gold objects $netacc^{\text{gold}}$. For the same input

data, we then run DSA without gold objects ($g = 0$) and measure its net accuracy $netacc^{\text{nogold}}$ (which is equal to its accuracy). The net accuracy gain is the difference between the net accuracy of DSA with gold and the net accuracy of DSA without gold

$$netaccgain = netacc^{\text{gold}} - netacc^{\text{nogold}}. \tag{4.9}$$

To find out which mean competencies benefit most from the use of gold strategies, we simulate the net accuracy gains as a function of the mean competence. As in the previous sections, we vary the mean competence $\bar{c}$ from $0 + w/2$ to $1 - w/2$ in 0.05 steps and set $w = 0.5$. For setting UNIFORM, we simulate the gold ratios $g \in \{0.05, 0.1, 0.15\}$. Our results indicate that SKEWED requires fewer gold objects than UNIFORM for similar gains. Consequently, for this setting, we simulate gold ratios $g \in \{0.02, 0.04, 0.06\}$.

Figure 4.7 presents the results of the simulation experiments. In particular, Figures 4.7a and 4.7b show the net accuracy gains of gold strategies for setting UNIFORM with 20 raters and 50 raters, respectively. Figure 4.7c shows the net accuracy gains of gold strategies for setting SKEWED.

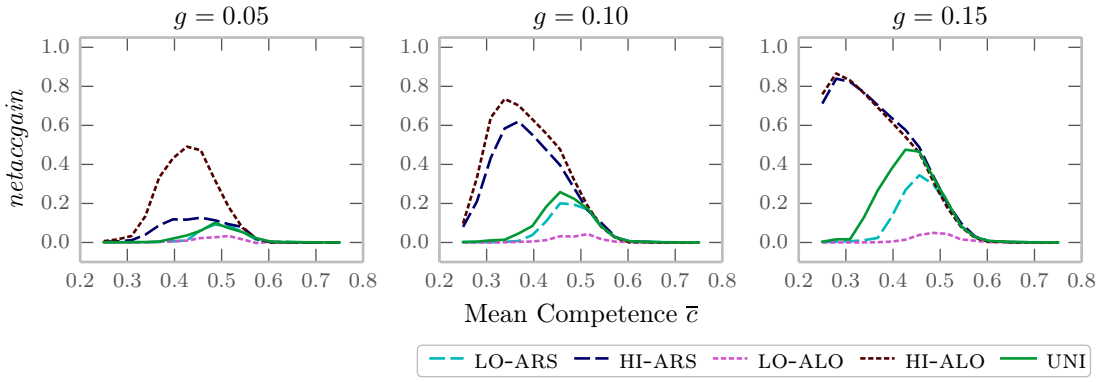
As we have expected, the net accuracy gains, both for UNIFORM and SKEWED, are higher for higher gold ratios. For all strategies, the gains for $\bar{c}$ greater than approximately 0.55 are close to zero. The reason is that the accuracy of DSA without gold objects is already high for $\bar{c} > 0.55$ (cf. Fig. 4.5) so there is not much room for improvement. For $\bar{c} < 0.55$, HI-ALO has gains greater than or equal to all other gold strategies. For some $\bar{c} < 0.55$, HI-ALO outperforms the other strategies by a wide margin. In the setting with a high number of ratings per data object (UNIFORM with 50 raters), the highest gains concentrate near $\bar{c} = 0.5$ for low gold ratios. In settings where ratings are sparse – either because the number of raters is low (UNIFORM with 20 raters) or because the rating rate is low (SKEWED) – and which have a high gold ratio, HI-ALO achieves high gains also for very low $\bar{c}$. In these settings, HI-ARS has a performance as good as or slightly worse than HI-ALO. We discuss the implications of these findings in Section 4.10.

4.7. Optimizing the Net Benefit of Gold Objects with an Adaptive Gold Algorithm

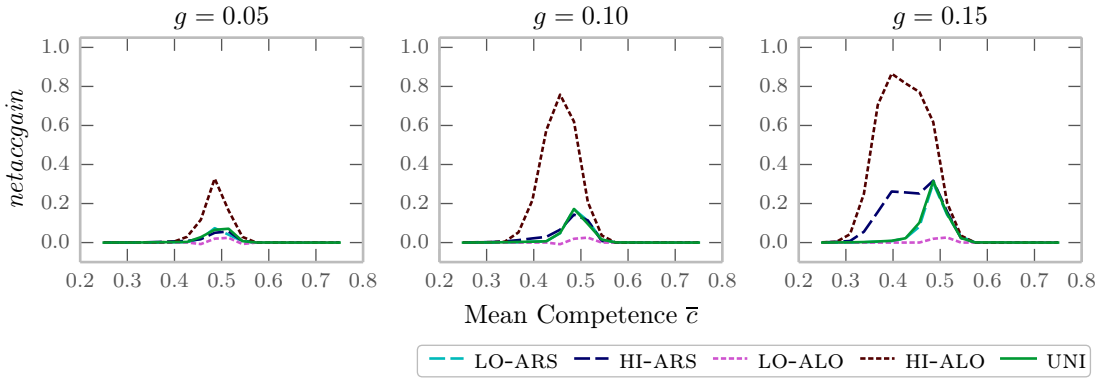
What is the optimal number of gold objects for DSA? This number not only depends on the benefits but also on the costs. As we have seen above, DSA benefits from gold objects by an increase in correctly classified non-gold data objects. This benefit is offset by the costs to obtain the gold objects. The exact costs and benefits depend on the specific scenario. For simplicity, we assume a benefit of 1 per correctly classified data object and costs of 1 per gold object. We define the net benefit as the difference of correctly classified non-gold data objects and the number of gold objects used by DSA

$$netbenefit = \sum_{k \in K \setminus K_{\text{gold}}} \mathbb{1}(\hat{o}_k = o_k) - |K_{\text{gold}}|.$$

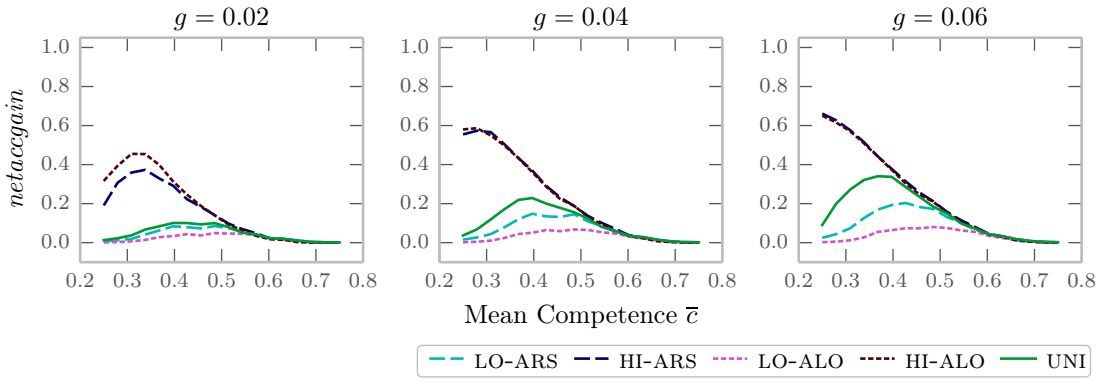
In the following, we discuss how to maximize the net benefit.



(a) Setting UNIFORM with 20 raters.



(b) Setting UNIFORM with 50 raters.



(c) Setting SKEWED.

Figure 4.7.: Net accuracy gains of DSA with different gold strategies using gold ratio g compared to DSA without gold objects ($g = 0$).

4.7.1. Adaptive Gold Algorithm

As before, we select gold objects by means of a gold strategy. But instead of fixing a gold ratio a priori, we let an algorithm decide how many gold objects to use in order to achieve the highest net benefit. The algorithm is adaptive, i.e., it decides when to stop based on runtime information. It works iteratively: Starting with zero gold objects, it adds one gold object per iteration. The goal of the algorithm is to stop adding further gold objects when the net benefit is highest. Of course, in the real world, the net benefit is unknown. Therefore, we cannot use it as a stop condition for the algorithm. Instead, we can only observe how the output of DSA changes in order to decide when to stop adding further gold objects. The adaptive gold algorithm is outlined in Algorithm 3. In

Algorithm 3: Adaptive Gold Algorithm for DSA.

Input: set of ratings $\{r_{i,k}\}$, gold strategy *gs*
Output: set of estimated types $\{\hat{o}_k\}_{k \in K}$, set of estimated response probabilities $\{\hat{P}(r_{i,k} = q \mid o_k = t)\}_{i \in I, q \in T, t \in T}$

$\{d_1, d_2, \ldots, d_{|K|}\} \leftarrow$ set of data objects ordered according to *gs*
$K_{\text{gold}} \leftarrow \emptyset$
$itr \leftarrow 0$
$\{\hat{o}_k^{itr}\}_{k \in K} \leftarrow$ run Algorithm 2 with $\{r_{i,k}\}$ and K_{gold} as input
repeat
  $itr \leftarrow itr + 1$
  select data object d_{itr} as gold object
  obtain expert ratings for d_{itr}
  $K_{\text{gold}} \leftarrow K_{\text{gold}} \cup d_{itr}$ (add d_{itr} to set of gold objects)
  $\{\hat{o}_k^{itr}\}_{k \in K} \leftarrow$ run Algorithm 2 with $\{r_{i,k}\}$ and K_{gold} as input
until *Eq. stop condition is satisfied* $\vee$ $itr \geq |K|$;

the following, we derive the stop condition.

Stop Condition Based on the Output of DSA

A stop condition that yields good results based on the DSA outputs is not obvious. This is because the changes of the output do not decrease monotonically over the iterations, as one might expect. Instead, they can vary strongly in either direction from one iteration to the next (cf. rightmost plot in Fig. 4.8). Further, they behave very differently in different simulations. Consider the *docchange* (data object classification change), i.e., the number of data objects whose classification has changed in iteration *itr* compared to iteration $itr - 1$ of the adaptive algorithm

$$docchange(itr) = \sum_k \mathbb{1}(\hat{o}_k^{itr-1} \neq \hat{o}_k^{itr}),$$

where $\hat{o}_k^{itr}$ denotes the estimated type of data object k in iteration *itr*, $itr \geq 1$. In the simulation run displayed in the leftmost plot in Fig. 4.8, the *docchange* does not

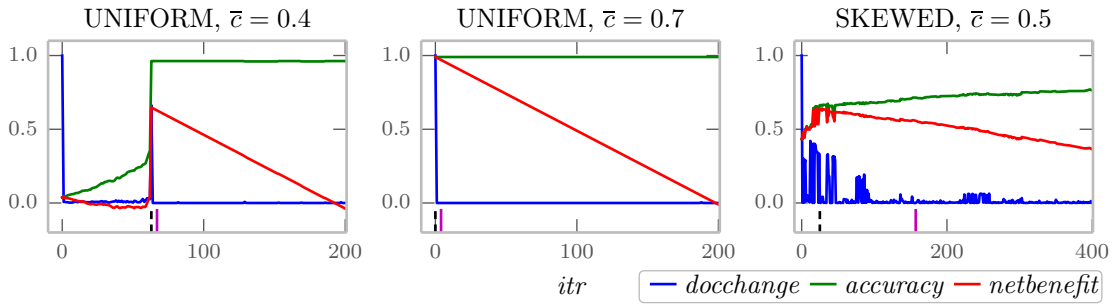


Figure 4.8.: Behavior of *docchange* (normalized), accuracy, and *netbenefit* over single runs of the adaptive algorithm. The vertical lines show the *itr* of the maximum *netbenefit* (black, dashed), and the *itr* where the stop condition with parameters $st_{\text{width}} = 4$, $st_{\text{maxsum}} = 2$ stops (magenta).

change much until iteration 63. Adding the 63rd gold object, however, gives DSA enough "knowledge" about the competence of the raters to reverse the classification of the data objects almost completely. This results in a large increase of the *netbenefit* and a large jump of the *docchange*. In the middle plot, the *docchange* is flat in the beginning as well, but there is no reversal of the classification later. In the rightmost plot, *docchange* changes with almost every iteration. The other outputs of DSA, e.g., the change of the competence estimates, behave similarly.

To gain robust results, our stop condition computes the sum of changes of the last st_{width} ("stop width") iterations. Let itr' be the current iteration. The stop condition tests if the sum of the *docchange* values of the previous st_{width} iterations is below the threshold st_{maxsum}

$$\sum_{itr=itr'-st_{\text{width}}}^{itr'} docchange(itr) \leq st_{\text{maxsum}}. \qquad \text{(stop condition)}$$

Finding Parameters for the Stop Condition

We have created a training dataset that contains the results of more than 1000 simulation runs of the adaptive algorithm to obtain values for st_{width} and st_{maxsum} that maximize the net benefit. For this training dataset, we have varied the rating distribution (uniform, Pareto with different parameters), the mean competence, the number of data objects, the number of raters, and the random seeds. Table 4.1 shows for each setting the combinations of st_{width} and st_{maxsum} that maximize the mean *netbenefit* over all simulation runs. We consider the HI-strategies only. This is because the LO-strategies and the UNI strategy perform strictly worse with the adaptive algorithm than the HI-strategies. Besides the training dataset, we have applied the adaptive algorithm to the test data we used in the previous evaluations of the gold strategies (cf. Section 4.6.2). We have computed the "net benefit to maximal net benefit ratio" *ntmnr*, i.e., the ratio of the *netbenefit* achieved by the st_{width}, st_{maxsum} combination and the maximum achievable *netbenefit* for each run

				mean *ntmnr*	
Setting	Gold Strategy	st_{width}	st_{maxsum}	training	test
UNIFORM	HI-ARS	3	0	0.96	0.95
UNIFORM	HI-ALO	1	0	0.98	0.98
SKEWED	HI-ARS	2	1	0.94	0.96
SKEWED	HI-ALO	2	1	0.89	0.94

Table 4.1.: Combinations of st_{width} and st_{maxsum} that maximize the mean *ntmnr* for different settings and gold strategies of the training dataset.

of the adaptive algorithm. Table 4.1 shows the mean *ntmnr* that the st_{width}, st_{maxsum} combination reached in the training dataset and when applied to the test dataset.

Further, the combination $st_{\text{width}} = 2$, $st_{\text{maxsum}} = 0$ yields *ntmnr* values almost as high as the values identified in Table 4.1. It is among the top-five combinations in each setting and each strategy tested. Using it might avoid overfitting to some degree, thus giving way to more robust results compared to the maximizing combinations when applied to other datasets.

4.7.2. Net Benefit Gains of the Adaptive Algorithm

We evaluate the adaptive gold algorithm by comparing its net benefit to the net benefit of the DSA without gold objects. As above, we consider the HI-strategies only. Similarly to Eq. 4.9, we define the net benefit gain as the normalized difference between the net benefit of the adaptive algorithm $netbenefit^{\text{adaptive}}$ and the net benefit of the vanilla DSA without gold $netbenefit^{\text{nogold}}$:

$$netbenefitgain = \frac{netbenefit^{\text{adaptive}} - netbenefit^{\text{nogold}}}{|K|}.$$

We evaluate the adaptive algorithm with the same simulation experiments detailed in Section 4.6.2. We use the robust parameters values $st_{\text{width}} = 2$ and $st_{\text{maxsum}} = 0$ for the stop condition of the adaptive algorithm. Figure 4.9 shows the net benefit gains and the gold ratio used. In general, the adaptive algorithm achieves very high gains for $\bar{c} \leq 5$. The HI-ALO strategy outperforms HI-ARS in the UNIFORM settings, while HI-ARS performs slightly better than HI-ALO in SKEWED. For each strategy in the UNIFORM settings, the adaptive algorithm uses a relatively high gold ratio for the lower competence range $\bar{c} < 0.5$. In this range, it also achieves the highest gains. In SKEWED, the adaptive algorithm uses a much lower gold ratio than in UNIFORM. Finally, for $\bar{c} > 0.6$ the gold ratio used by the adaptive algorithm is close to zero in all settings and for both gold strategies. The net gain in this range is zero or slightly negative.

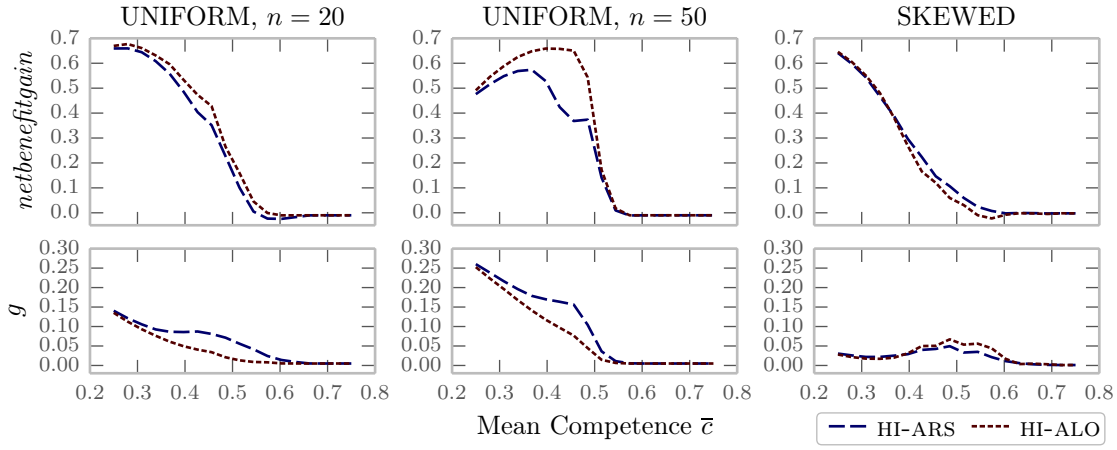


Figure 4.9.: Net benefit gains and gold ratio g used by adaptive gold algorithm compared to DSA without gold objects. Stop condition with parameters $st_{\text{width}} = 2$ and $st_{\text{maxsum}} = 0$.

4.8. Using Gold Strategies to Counter Collusion Attacks against the Dawid-Skene Algorithm

The previous discussion focused on improving the accuracy and the net benefit of DSA in particular in the presence of low-competence raters. A potential problem that has a negative impact on the accuracy of DSA as well are collusion attacks. In a collusion attack, raters coordinate to rate the same data objects with the same value to artificially increase their estimated competence. This is beneficial for the colluders if they receive a remuneration for their ratings that is based on their estimated competence. For example, many online communities remunerate users with reputation or Karma points for high-quality contributions. We propose to compute remunerations in such communities contingent on the competence estimates by DSA. Similarly, Wang et al. propose an algorithm that pays crowdsourcing workers based on, among other metrics, the competence estimates calculated by DSA [WIP]. (However, as mentioned, they do not address collusion attacks.) In such settings a collusion attack allows colluders to artificially increase their remuneration while saving cognitive effort for determining the truthful value of the data objects. Since DSA assigns an inflated weight to their low-competence ratings colluders can also severely damage the accuracy of DSA.

Gold objects can counter a collusion attack. They allow for more precise competence estimates. Thus, they correct the overly high competence estimates of colluders. This reduces the benefit gained by colluders, thereby making collusions less desirable. It also reduces the damage of collusions on the accuracy of DSA.

4.8.1. Model of a Collusion Attack

We extend the model of a peer-rating online community introduced in Section 4.1.

Our model of a collusion attack partitions the set of raters into a set of colluders $I_{\text{col}} \subseteq I$ and a set of *honest raters*, i.e., raters that do not collude, $I_{\text{hon}} = I \setminus I_{\text{col}}$. Colluders coordinate – for example by using the internet as a communication channel – to give the same ratings for each data object in a subset of the data objects. We call the subset of data objects that colluders use for the collusion attack *collusion data objects* and denote it K_{col}. For simplicity, we assume that every colluder rates all data objects from the set K_{col} but no other data objects. The set of non-collusion data objects $K_{\text{hon}} = K \setminus K_{\text{col}}$ is the set of data objects that colluders do not rate. (Honest raters rate objects from both K_{hon} and K_{col}.) Without loss of generality, we assume that colluders always assign ratings with value 1 independent of the true type of the data object in question. I.e., $r_{i,k} = 1$ for all $i \in I_{\text{col}}$ and for all $k \in K_{\text{col}}$. Other collusion strategies – like coordinating on a rating value per data object or, if the prior is highly skewed, choosing the a priori most likely value – add little to the discussion at hand (we argue).

We use $n_{\text{col}} = |I_{\text{col}}|$ and $m_{\text{col}} = |K_{\text{col}}|$ to denote the number of colluders and the number of collusion data objecs, respectively. We define the *ratio of collusion objects* as the ratio of the number of collusion data objects to that of all data objects, i.e., m_{col}/m. Further, we define the *ratio of colluders* as the proportion of colluders among all raters, i.e., n_{col}/n.

To simplify the discussion, we assume type-independent competencies c_i of raters and colluders. We use the *collusion rent* as a metric for the benefit raters gain from colluding. The collusion rent of a colluder i is the difference between his estimated competence and his real competence $\hat{c}_i - c_i$.

Since we assume that all ratings of colluders are 1, their competence equals the prior probability of type 1, that is, $c_i = p(1)$ for all colluders $i \in I_{\text{col}}$. In other words, collusion ratings are correct with probability $p(1)$. From the definition of the colluders rent it follows that the maximum collusion rent is $1 - p(1)$.

4.8.2. Influence of Colluders and Honest Raters on the Outcome of a Collusion Attack

Honest ratings, i.e., the ratings of honest raters, are a countermeasure against collusions. A high number of honest ratings for a given data object makes it less likely for colluders to influence the classification of the data object.

We conduct a simulation experiment to gain intuition on how the number of honest ratings and the number of collusion ratings influence the outcome of a collusion attack. In particular, we focus on the number of both honest and collusion ratings per collusion data object. In the simulation, we vary three parameters: (1) the number of colluders n_{col}, (2) the number of honest raters n_{hon}, and (3) the rating rate of honest raters rr_{hon}. All three parameters determine the ratio of honest ratings to collusion ratings per collusion data object. The parameter rr_{hon} is not an exogenous parameter of SKEWED, i.e., we can only indirectly manipulate it by changing the parameters of the Pareto rating distribution. This is why we use the setting UNIFORM for this simulation only. Our intention behind the simulation at hand is solely to build intuition. Therefore, we do not deem the omission of SKEWED limiting. We simulate both settings below where

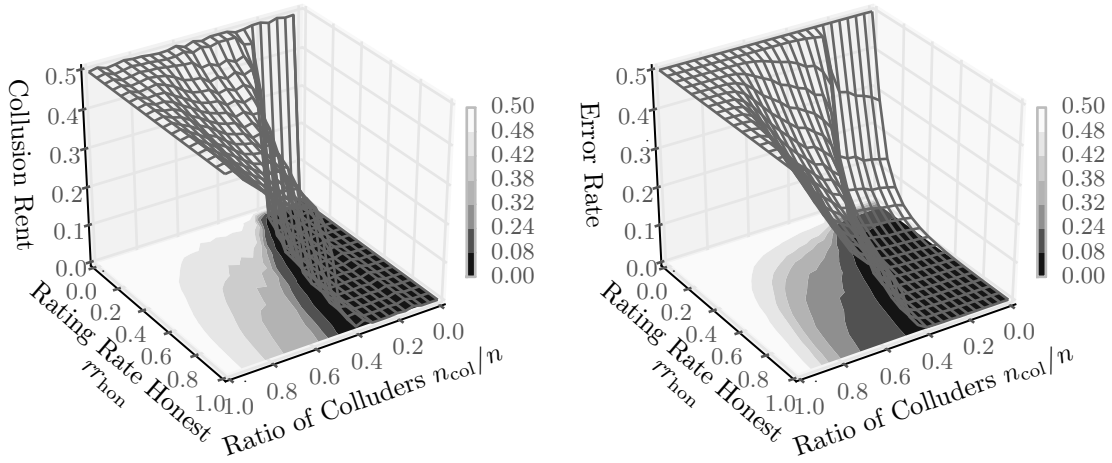


Figure 4.10.: Average collusion rent and error rate of DSA as functions of the rating rate of the honest raters and the ratio of colluders.

we evaluate how gold strategies can reduce collusion attacks. In the simulation, we vary rr_{hon} from 0 to 1 in 0.05 steps and assign each honest rater $i \in I_{\text{hon}}$ the same rating rate $rr_i = rr_{\text{hon}}$ (see Section 4.4 for the definition of rr_i). We fix the number of raters to $n = 50$ and vary the ratio of colluders n_{col}/n from 0 to 1 in 0.04 steps. Note that this varies both the number of colluders n_{col} and the number of honest raters n_{hon}. We set the ratio of collusion objects to $m_{\text{col}}/m = 0.3$, i.e., colluders coordinate on 30 percent of the data objects. The simulation draws the competence of the honest raters from a uniform distribution on the interval $[0.35, 0.95]$.

Figure 4.10 shows the average collusion rent (left-hand side) and the error rate of DSA (right-hand side) resulting from a collusion attack. Both collusion rent and error rate are shown as a function of the rating rate of the honest raters rr_{hon} and of the ratio of colluders among all raters n_{col}/n. As expected, a collusion attack requires more colluders to maximize the collusion rent the higher the rating rate of honest raters.

Once the ratio of collusion ratings to honest ratings (determined by the combination of rr_{hon} and n_{col}/n) reaches a certain threshold, the collusion rent rises sharply. For example, in the left-hand plot, this threshold runs approximately from (0, 0.2) to (1, 0.4) through the rr_{hon}-n_{col}/n plane. Intuitively, if there are enough collusion ratings for a data object $k \in K_{\text{col}}$, DSA shifts the type estimate of k in favor of the collusion ratings. Because of this, DSA assigns higher competence estimates and thus higher weights to the colluders. The higher weights increase the influence of the collusion ratings on the other collusion data objects, and so on.

Effect of Collusions on the Error Rate of Non-Collusion Data Objects

The effect of collusions on the error rate is two-fold: they directly affect the error rate of collusion data objects and indirectly affect the error rate of non-collusion data

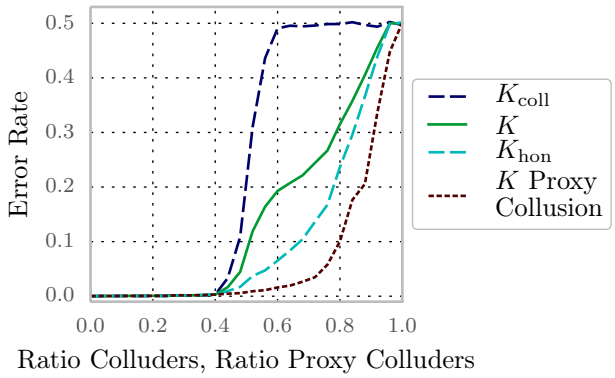


Figure 4.11.: Error rate of different subsets of data objects for colluders and proxy colluders.

objects. The indirect effect is as follows. Colluders decrease DSA's accuracy of the type estimates of collusion data objects. This in turn lowers DSA's accuracy of the competence estimates of those honest raters who have also rated the collusion data objects. The lowered accuracy of the competence estimates of honest raters decreases the accuracy of non-collusion data objects.

Figure 4.11 represents the two-dimensional slice of Figure 4.10 where the rating rate of honest raters is 1.0. It differentiates between the error rates of three different subsets of data objects – of all data objects (K), of collusion data objects (K_{col}), and of non-collusion data objects (K_{hon}). The error rates are shown as a function of the ratio of colluders.

To visualize the indirect influence of colluders on the error rate of K_{hon} we simulate a *proxy collusion*. For this purpose, we run the simulation experiment of a collusion attack described above again and replace the colluders with *proxy colluders*. Proxy colluders are regular raters, i.e., they do not coordinate. To make them comparable to colluders, they rate the same number of data objects as the colluders do – but not necessarily from the set K_{hon} only. Further, they have the same observed error rate as colluders, i.e., they rate as many data objects incorrectly as colluders. The other simulation parameters are the same as before.

Figure 4.11 shows the error rate for all data objects K in a proxy collusion as a function of the ratio of proxy colluders. Note that there are no equivalents to K_{hon} and K_{col} in a proxy collusion. This is because proxy colluders do not coordinate on a subset of K. For the collusion attack, the error rates of K, K_{col}, and K_{hon} are all higher than the error rate of K in a proxy collusion. Just by coordinating, colluders cause higher error rates – even for the data objects K_{hon} they did not rate – than the otherwise identical proxy colluders.

4.8.3. Reducing the Collusion Rent with Gold Objects

We show that gold objects can reduce the collusion rent and the error rate which result from a collusion attack. As an example, we simulate the setting discussed in the previous

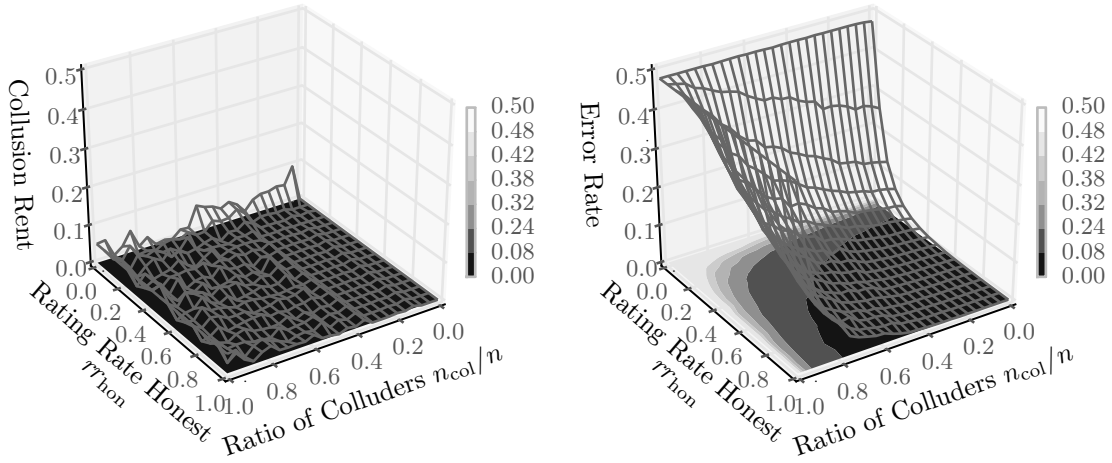


Figure 4.12.: Colluders rent and the error rate of DSA with five percent gold objects. Otherwise same setting as in Figure 4.10.

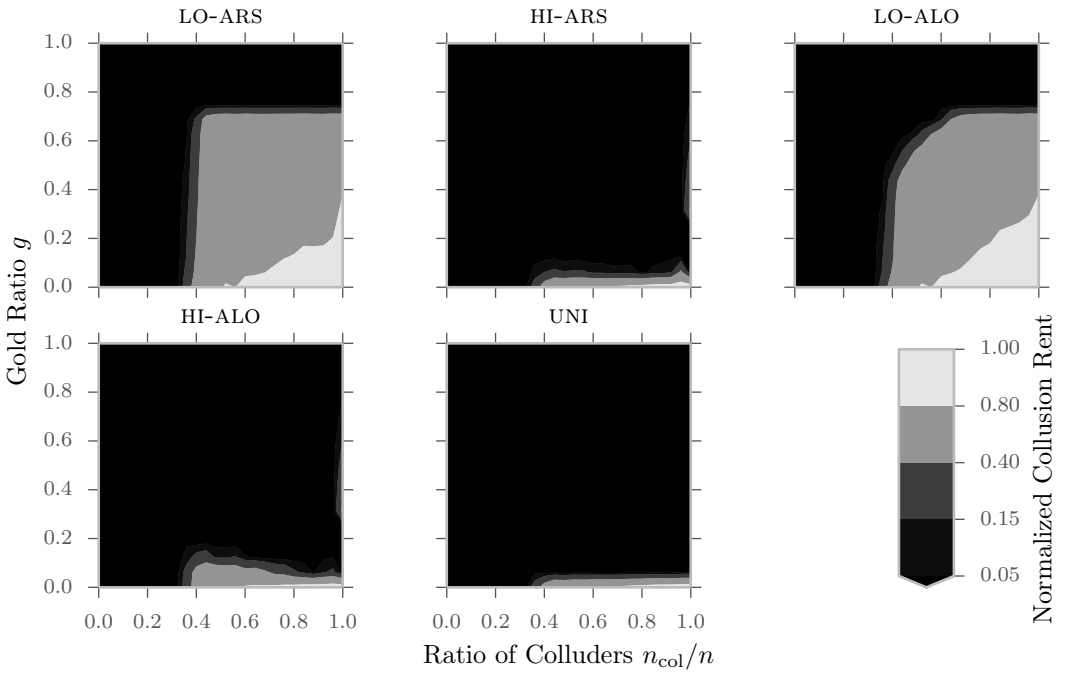
section but using five percent randomly chosen data objects as gold objects. This (see results in Figure 4.12) reduces the collusion rent to almost zero. It also reduces the error rate significantly compared to the collusion attack without gold objects (cf. Figure 4.10).

In the following, we analyze the influence of gold strategies as a means to reduce the collusion rent. The analysis focuses on reducing the collusion rent for the following reason. We assume that raters consider the collusion rent an incentive for colluding. Therefore, a reduced collusion rent makes colluding less desirable and thus should lower the occurrence of collusions.

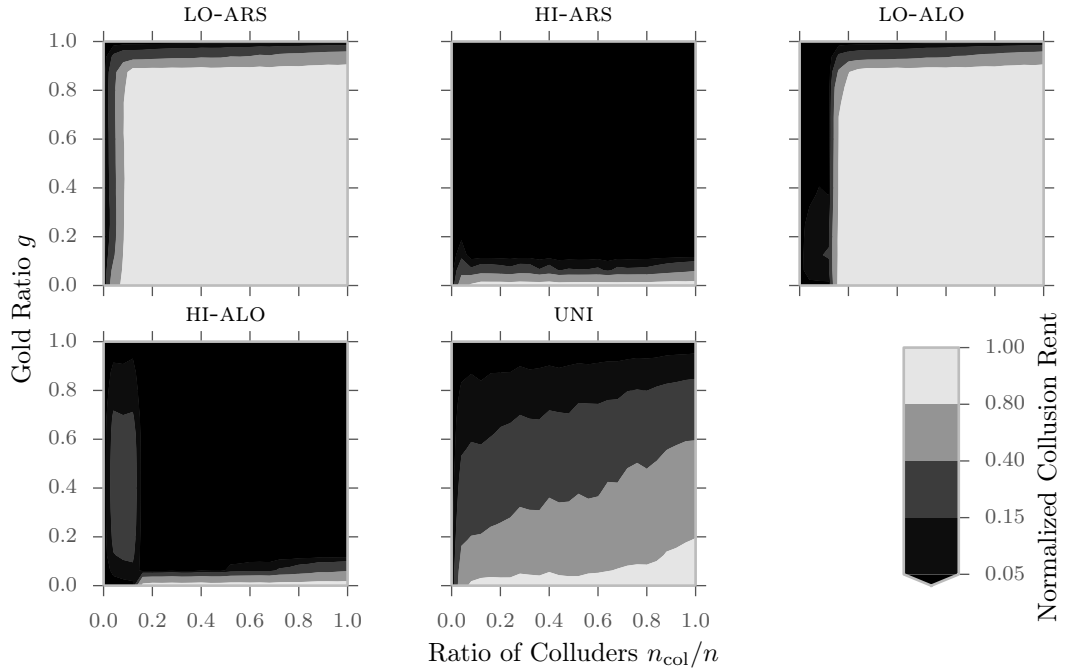
To investigate the effects of gold strategies on the collusion rent, we conduct several simulation experiments for UNIFORM and SKEWED. In these experiments we vary the gold ratio g and the ratio of colluders n_{col}/n. We fix the other simulation parameters of UNIFORM and SKEWED to their default values defined in Section 4.4. As in the previous section, we set the ratio of collusion objects m_{col}/m for the setting UNIFORM to 0.3. In setting SKEWED, honest raters rate approximately 2.6 percent of the data objects. This is a much lower rating rate than in setting UNIFORM where honest raters rate 40 percent of the data objects. Accordingly, we adjust the ratio of collusion data objects for SKEWED to a lower value as well and set it to $m_{\text{col}}/m = 0.1$, i.e., colluders coordinate on 10 percent of the data objects.

As we have seen in the previous section, the maximum achievable collusion rent depends on the type prior $p(t)$. The type priors of UNIFORM and SKEWED differ. To make the results comparable between the two settings, we calculate the *normalized collusion rent*. The normalized collusion rent for a simulation is the collusion rent divided by the maximum collusion rent observed in that simulation.

Figure 4.13 shows the normalized collusion rent (0/black: best, 1/white: worst outcome) as a function of the colluders ratio and the gold ratio, for the different gold strategies. For SKEWED (Figure 4.13b), HI-ARS pushes the collusion rent to 0 for gold ratios larger



(a) Setting UNIFORM with ratio of collusion objects 0.3.



(b) Setting SKEWED with ratio of collusion objects 0.1.

Figure 4.13.: Collusion rent for different gold strategies as a function of the ratio of colluders and the gold ratio.

than 0.1 independent of the ratio of colluders. HI-ALO reduces the collusion rent to almost 0 for most of the area of the contourplot as well. The LO strategies, on the other hand, can only reduce the collusion rent to close to 0 for very high gold ratios (> 0.95) or very low colluder ratios (< 0.02 and < 0.08 for LO-ARS and LO-ALO, respectively). The strategy UNI performs better than the LO strategies but worse than the HI strategies in this setting.

For UNIFORM the differences between HI and LO strategies are lower but still pronounced (Figure 4.13a). Here, HI strategies perform better than LO strategies as well. Colluders in this setting gain rent > 0.05 only if their ratio is larger than 30 percent, independent of the gold strategy. The UNI strategy performs best in this setting. It reduces the collusion rent to 0 for gold ratios > 0.05. HI-ARS performs only slightly worse than UNI.

The reason why the HI strategies are so effective in reducing the effects of a collusion attack is that they select data objects as gold objects that have a high level of agreement. A high level of agreement is a property of collusion data objects since all colluders agree on those data objects. I.e., HI strategies are a collusion detection mechanism.

4.9. Related Work

There exists a large body of literature on the accuracy of MV. The earliest work on the accuracy of MV, the Condorcet jury theorem, dates back to 1785 [Con85]. Grofman et al. review results on the accuracy of voting processes as a function of the competences of the individual voters, the decision procedure, and the number of voters [GOF83]. The optimal weights for raters with type-independent competence in binary choice situations (Equation (4.6)) have been identified several times independently [MP69, DH73, NP82]. Lam et al. generalize the Condorcet jury theorem to cases where even numbers of voters are allowed, where the number of choices is greater than two, and thus ties are possible [LS97]. Kuncheva et al. derive upper and lower limits on the accuracy of MV for both dependent and independent classifiers [KWS03].

Li et al. derive theoretical error bounds (approximate and expected) on linear threshold rules, such as MV and WMV, for binary labeling tasks [LYZ13]. This includes bounds on the expected error rate of the MAP rule with known competencies.

[DS79] have developed the Dawid and Skene algorithm for the estimation of error rates made by clinicians in the assessment of patients. The algorithm is based on the general algorithm for expectation maximization (EM) developed by Dempster et al. [DLR77]. Whitehill et al. introduce an EM algorithm similar to DSA that takes varying difficulties between the data objects of the same type into account [WRW$^+$09].

Snow et al. discuss an experiment for the estimation accuracies of AMT workers for word annotation tasks [SOJN08]. Similarly to DSA, their method estimates the worker competencies based on comparisons with gold standard examples, and uses a MAP estimate to infer the annotation quality.

[IPW10] develops an algorithm based on so-called soft labels and expected costs of each soft label to differentiate between biased raters ($c < 0.5$) and spammers ($c = 0.5$).

The algorithm uses DSA to estimate rater competencies. In an experiment, the algorithm performs better at detecting those spammers that always rate the class with the highest type prior. This leads to a higher accuracy (0.998) compared to DSA (0.95). The algorithm assumes that DSA returns "some reasonably accurate estimates" of the competencies. However, this assumption does not hold true for $\bar{c} < 0.5$, as our results show.

Similarly, Raykar et al. [RY12] propose an algorithm to eliminate spammers and malicious raters. They compare their algorithm to MV and DSA and find that it has a better area under the ROC curve than MV and DSA, in particular if the fraction of spammers is high. Further, they find that their algorithm is better at detecting spammers than MV and DSA. However, their algorithm (implicitly) assumes that the mean competence $\bar{c}$ is higher than 0.5. For example they state that "the methods proposed in Dawid and Skene [...] can automatically flip the labels for the malicious annotators", which is clearly not true if $\bar{c} < 0.5$ as our results show. Further, it is clear from the description of their simulation experiments that the settings they study to show the effects of their method have a mean competence greater than 0.5.

Wang et al. [WIP11] propose an integration of gold objects into DSA. Further, they develop an algorithm that integrates gold objects in a setting with Amazon Mechanical Turk workers, as opposed to an open community setting like ours. Their method tells AMT workers to label a priori created gold objects based on the expected utility of additional ratings by these workers. However, forcing members of the community to rate particular contributions is not possible in our setting.

4.10. Discussion

To discuss the implications of our findings, we distinguish between four situations where a community of raters classifies data objects.

The first situation is a typical crowdsourcing setting with payments such as Amazon Mechanical Turk that lets the employer decide which data objects a crowd worker has to rate. There, the method proposed by [WIP11] might allow using gold objects efficiently based on the expected utility of additional ratings issued by the crowd worker.

The second and the third situation occur in an open community setting where it is not possible to force raters to rate specific data objects. Here, we have to differentiate between communities with a high probability that the mean competence is greater than 0.5 and those with the risk of having a mean competence less than 0.5. If there is a high probability that the mean competence is greater than 0.5, DSA will classify data objects with high accuracy. Our results show that gold objects cannot improve this situation much.

If, on the other hand, the open community faces the risk that its mean competence is near 0.5 or lower, DSA or related methods will not suffice. Such a low mean competence occurs whenever the community is afflicted by a large fraction of low-competence raters, such as spammers, biased raters, malicious raters, and raters with a consistent misunderstanding. In that case, the gold strategies we have proposed can increase the classification accuracy considerably, compared to the vanilla DSA without gold objects. Here, the

preferred gold strategies are HI-ALO and, to a lesser degree, HI-ARS. The accuracy gains achieved by a given gold ratio depend on the characteristics of the community. Roughly, the more homogeneous the community, the higher the rating rate, and the lower the competence, the higher the gold ratio required to achieve high accuracy gains compared to the vanilla DSA. In case of a typical open online community with sparse ratings and a skewed rating distribution, a gold ratio as low as two percent can be enough to offset the impact even of a large number of spammers and biased raters. Further, if we also take the costs of gold objects into account, the adaptive gold algorithm can determine the optimal gold ratio with high accuracy.

As the fourth situation, communities face the risk of collusion attacks, in particular if raters are remunerated based on their competence inferred by DSA. Here, gold strategies can successfully counter collusion attacks. HI-ARS is particularly effective against collusion attacks, both in homogeneous settings and in settings with a skewed rating rate that is typical for open communities.

In summary, HI-ARS and HI-ALO can safeguard against the risk of both the impact of (i) a large fraction of low-competence raters, and (ii) of collusion attacks. In such circumstances, a low ratio of gold objects together with the HI-ALO and HI-ARS strategies can be much more effective than simply selecting gold objects randomly. Further, the optimal gold ratio does not need to be guessed but can be determined with high accuracy by our adaptive gold algorithm.

4.11. Conclusion

In this chapter we have analyzed the problem of classifying contributions in an open peer-rating online community. We have discussed the accuracy of majority voting schemes under homogeneous competencies and known heterogeneous competencies. We have analyzed the estimation quality of DSA in various settings. We find that in a homogeneous setting, where raters have roughly the same, relatively high, rating rate, DSA stabilizes after it has classified a relatively low number of data objects. In a more open setting, on the other hand, with a low rating rate and a skewed rating distribution that is typical for open internet communities, DSA requires a much higher number of data objects to stabilize. Further, we find that for a mean competence higher/lower than 0.5, DSA performs better/worse than majority vote. This effect becomes more pronounced, the more widespread the competence distribution is, and the higher the number of raters is.

We have proposed and tested gold strategies based on the level of agreement to increase the accuracy of DSA in low-competence settings. Further, we have proposed and evaluated an adaptive algorithm to maximize the net benefit of gold objects. Finally, we have discussed the damage done by collusion attacks against DSA and have tested how gold strategies can reduce this damage. A main finding of ours is that the HI-ALO gold strategy is very effective in increasing the accuracy of DSA in low-competence settings. Further, the adaptive algorithm determines the optimal gold ratio for each strategy and each setting with high accuracy. Finally, we find that the HI-ARS and the UNI strategy are effective in reducing the benefit gained by colluders. Thus, they render collusions

less attractive for raters. We have discussed the implication of these findings and have found that HI-ALO and HI-ARS can effectively safeguard typical open online communities against the risk of both the impact of (i) a large fraction of low-competence raters, and (ii) of collusion attacks. In such circumstances, a low ratio of gold objects together with the HI-ALO or HI-ARS strategy can be much more effective than simply selecting gold objects at random.

We have focused on a binary setting in this chapter. However, the methods we have discussed are also applicable to multi-type settings, i.e., $|T| > 2$. In particular the gold strategies can be readily modified to work in multi-type settings. In this case, ars_k would have to be computed for each type t of a given data object individually. For this computation each rating needs to be encoded as 1 if it is in favor of t, as -1 otherwise. The modification of strategies based on alo_k is straightforward as well: instead of using alo_k the modified strategies select data objects that have the highest/lowest MAP estimate for any of their types.

5. Reviewing the Reviewers: a Study of Author Perception on Peer Reviews

One of the most long-standing models of peer production is that of the scientific community [Hay09]. To meet the problem of quality assurance and assessment in peer production (Chapter 1), science uses peer review.

Perhaps the first description of a peer-review process can be found in the book *Ethics of the Physician* by Ishāq ibn 'Alī al-Ruhāwī (CE 854–931) of Al Raha, Syria [Spi02]. The book states that it is the duty of a physician to take notes about a patient on each visit. After the patient had been cured or had died, a local council of physicians examines the notes. Based on their judgment, for example, a maltreated patient or the patient's relatives (in case of death) could sue the practicing physician for damages: "[...] the physician meets with the experienced people, he brings out the record to be examined by knowledgeable professionals in medicine. If the disease proves to be the same as was told, and the signs were the signs of the disease that are characteristic for it, and the drugs and management were satisfactory, the physician would be thanked and would leave. If not, he shall get what he deserves" [AK97].

Peer review was introduced into the field of science in 1731 by the Royal Society of Edinburgh that published a collection of peer-reviewed medical articles [Kro90]. The broad adaption of the peer-review process in the sciences occurred in the middle of the twentieth century, presumably because at that time (a) the number of articles increased strongly and so did the competition for publication space in scientific journals, and (b) the availability of photocopiers made replication for the purpose of reviewing feasible [Spi02].

Today, peer review is the de facto standard for ensuring quality in science. Reviewing a scientific paper requires considerable intellectual effort and time. However, the incentive to write high-quality reviews tends to be somewhat low, as reviewers are not remunerated for their efforts [BBC+07] and usually remain anonymous. While most reviewers do provide high-quality reviews, there is a non-negligible rate of reviews of lower quality, at least according to the perception of the authors. Personal communication with other scientists as well as numerous discussions regarding the pros and cons of peer reviewing in various scientific communities show this [CKG02, Nat06a, BBC+07].

We believe that feedback given by authors has potential to improve the review process. More specifically, we deem it promising to rely on review ratings to identify high-quality reviews and remunerate reviewers.[1] However, the specifics of such a remuneration mechanism are not obvious. For instance, assuming that accept/reject decisions affect the perception of authors, simply remunerating reviewers based on the ratings they receive

[1] The form of the remuneration is not a topic of this chapter. One possibility is to remunerate reviewers with specific awards, e.g., 'best reviewer award', as some conferences have done already.

from authors is not objective. To illustrate: A review of a rejected paper is likely to obtain low ratings. Had the same paper been accepted, on the other hand, the review would presumably receive higher ratings, should that assumption hold.

To gain insight into authors' perception of reviews, we have conducted a study with authors who had submitted papers to a peer-reviewed computer science conference. One important goal was to determine which criteria may be useful to identify high-quality reviews and thus to determine an adequate basis for reviewer remuneration. Based on our study, one might be able to derive other measures as well, e.g., re-design of review forms, or other measures which we have not come up with at this current point of time. In this thesis, we keep the discussion focused on reviewer remuneration as the core objective. Moreover, our study addresses the questions: How is author satisfaction with review quality distributed? How strongly do the characteristics of the review, in particular the review scores, as well as the accept/reject decision, affect author ratings? Which of the different assessments of the reviewer influence author perception of overall review quality?

To this end, we incorporated review ratings into the review process. Authors could assess each review they had received according to a broad selection of criteria, such as helpfulness of review comments. We have also asked them to rate the review scores they had received.

For the sake of clarity, we distinguish assessments by reviewers and those by authors. We refer to values used for assessment with the following two terms:

- *scores*, when issued by reviewers for the articles, and
- *ratings*, when issued by authors for the reviews.

Further, we deem HRMs (Section 2.2) a promising means to reward reviewers based on the assessments of other reviewers of the same submission. HRMs rely on the assumption that respondents use their own opinion as evidence for the popularity of this opinion among others (see 'Empirical evidence for Bayesian updating' on page 18). We test whether this assumption holds in a scholarly peer-review setting.

Given our results, we discuss how a suitable metric to remunerate reviewers could look like. This metric should neutralize possible effects of the review process, e.g., the effects of review scores, on author ratings as much as possible.

We find that the authors' satisfaction with review quality is good, though improvable. We also find that authors rate reviews as good if they deem the review helpful, if they deem the review comments justified, and if they have the impression that the reviewer made an effort to understand the paper. Acceptance status and self-assessed reviewer expertise only have a weak influence on perceived review quality. Unsurprisingly, authors assess reviews more favorably if they assign high scores. We design a remuneration function for reviews based on ratings and scores that, when applied to the data collected in our study, neutralizes the effects of scores on the remuneration to a large degree.

The remainder of this chapter proceeds as follows. Section 5.1 reviews related work. Section 5.2 presents the questionnaire used for the study, its implementation and the statistical methods we use for the analysis. Section 5.3 present the results of the analysis.

Section 5.4 discusses the results. Section 5.5 studies the suitability of ratings as a basis for remuneration of reviewers. Section 5.6 concludes.

5.1. Related Work

Criticism of peer reviewing has concentrated mainly on its efficacy and effectiveness. Some studies [WKWea02, G+08, G+90] have surveyed authors who had submitted manuscripts to journals. However, the results from the surveys differ from each other. Gibson et al. report on an online survey of 445 authors of research manuscripts submitted to the *Obstetrics and Gynecology* journal [G+08]. Authors were asked to rate six aspects of editorial comments and three aspects of the review process. Further, they let the journal's senior editors rate the reviews as well. One result is that authors' ratings did not correlate with ratings of reviews by the journal's senior editors. Further, they find that the authors of accepted manuscripts give higher ratings for overall satisfaction than authors of rejected manuscripts. Garfunkel et al. find a weaker correlation between author ratings and manuscript fate [G+90]. Gibson argues that the difference (between Garfunkel et al. and his own findings) results from the number of survey items and the rating scales in the questions.

We see many exogenous factors which might influence author satisfaction with peer reviewing, for instance the organization of the review process, the selection of reviewers and the design of the review forms. In addition, the review process of a conference is different from the one of a journal. As [SB94] has pointed out, at least for experimentalists, conference publication is preferred to journal publication. Moreover, the premier conferences tend to be more selective than the premier journals. Hence, many conferences have huge numbers of submissions combined with tight time constraints. Publication in conferences needs shorter time to print (7 months vs. 1-2 years). However, there is a lack of studies on conference reviews.

There also are various proposals to increase review quality. Some proposals attempt to improve the review process itself, like allowing authors to submit feedback in the rebuttal phase or supporting a rather open review process instead of double blind. In the journal *Biology Direct* [Nat], to give an example, authors can select their reviewers from the editorial board, and reviews are not only signed, but also published together with author responses as part of each article. Analyses of different modes of peer-review activities, e.g., online vs. face-to-face reviewing [B+09], exist as well. Van Rooyen et al. study open peer review [VRGE+99]. They find that asking reviewers to reveal their identity to authors had no important effect on the quality of the review, and on the recommendation regarding publication. However, it significantly increased the likelihood of reviewers declining to review.

Others have proposed to train reviewers. *The British Medical Journal* offers reviewers a workshop which gives them clear briefs, including guidance on what to include in the review etc. [Nat06b]. Callaham et al. try to improve reviewing skills by means of feedback from the editorial board. In their study editors write short feedback in text to the reviewers to comment on the quality of the reviews submitted [CKG02]. However, the

performance of reviewers is hardly improved, i.e., simple written feedback to reviewers seems to be inefficient as an educational means in this specific context. Another study finds that reviewer ratings given by journal editors are moderately reliable, and that they correlate modestly with the ability of reviewers to find flaws in a test manuscript [CBWW98].

Peer reviewing not only is an important instrument in the scientific community to pick good contributions, but also finds its usage in other disciplines. In software-engineering processes, to give an example, peer reviews are used to detect deficiencies in the code [Gal04, Wie02]. Other studies investigate the effect of peer reviewing on student learning. In [B+09], students review papers written by their peers, and the results indicate that students take peer reviews seriously and provide constructive reviews.

5.2. Materials and Methods

We have carried out our survey by means of an online questionnaire. Survey participants were the authors of the CASES 2009 conference. In this section, we first describe details of the conference and its peer-review process which are relevant to our study. Then we describe the questionnaire and the implementation of the study. Finally, we list the statistical methods we use in our analysis.

5.2.1. Conference and Peer-Review Process

We invited the authors of the *CASES 2009 Conference for Emerging Technology in Embedded Computing Systems* to participate in our study. The conference is held annually and focuses on compilers and architectures for embedded systems [CAS09]. Authors submitted 72 papers to the CASES 2009 conference overall. In all, 48 reviewers wrote 311 reviews on the submissions. The number of reviewers per submission ranged from 2 to 6 (avg.= 4.38 reviewers/submission). Out of the 72 submissions, the conference rejected 47 and accepted 23 as full papers and 2 as short papers. The reviewers had not been aware of our study beforehand.

The reviewers had to assign the following five detail scores: *Originality*, *Technical Contribution*, *Experimental Results*, *Description of Related Work*, and *Language and Clarity*. To assess the overall quality of the submission, each review contained one *Overall Score*. Additionally, reviewers provided a numerical self-assessment of their own expertise regarding the topic of the submission. Scores and self-assessment were based on the usual 1-5 scale, with 1 being the minimum and 5 the maximum score. In addition, reviewers could provide written comments. The conference chairs based their accept/reject decisions mainly on the *Overall Score*. However, they revised some of the ranking-based decisions during a one day face-to-face meeting.

5.2.2. Questionnaire

Immediately after the authors had received their notification of acceptance/rejection, we invited them to fill out a questionnaire. The questionnaire consisted of two parts. The first

part contained questions concerning each individual review the respective submission had received. The second part contained general questions. The response formats were mostly ordinal and differed depending on the question. Some questions elicited interval-level data. See Appendix D for an offline version of the questionnaire.

Survey Rating for	# Choices	Choices
Overall Quality	5	'very low' to 'very high'
Approp. of each review score	3	'too low', 'appropriate', 'too high'
Perceived expertise of reviewer	5	1-5
Helpfulness for future work	4	'not at all' to 'very helpful'
Approp. of review length	3	'too short', 'appropriate', 'too long'
Effort of reviewer	3	'low', 'average', 'high'
Percent of justified comments	5	0%, 25%, .., 100%

Table 5.1.: Review specific ratings and response formats.

Part I – Review specific ratings. Regarding the individual reviews for a submission, the following assessments were part of our questionnaire (see also Table 5.1):

- *Overall Quality.* We asked to the authors to assess the overall quality of the review ("What is your overall rating of the quality of Review?").
- *Appropriateness of each of the six review scores.* We elicited ratings regarding the appropriateness of the five detail scores, as well as of the *Overall Score*.
- *Perceived expertise of reviewer.* We also let the authors rate the expertise level of the reviewer on the same scale as the reviewers' self-assessment.
- *Further criteria that might influence review quality.* Additionally, we asked questions addressing the criteria which might influence review quality: helpfulness of the review comments for future work, appropriateness of review length, perceived effort of the reviewer to understand the paper, percentage of justified comments.

Part II – General questions. To test whether authors act in line with Bayesian updating, we let them estimate the ratio of reviews rated 'very low' or 'low' among (i) all authors, (ii) authors whose submissions had been accepted, and (iii) authors whose submissions had been rejected. Finally, we asked authors whether they deem ratings likely to improve review quality.

5.2.3. Implementation of the Survey

As mentioned, we sent out invitations to participate in the survey immediately after the notifications. We invited the contact author, i.e., one author per submission. We did not invite multiple authors per submission to avoid that authors distort results by answering

questionnaires for their co-authors. Moreover, we assume the opinions of co-authors to be highly correlated.

Recall that the number of reviews per submission varied from two to six. We set up the questionnaire software so that the number of questionnaire items matched the numbers of reviews a submission had received. For example, if a submission x had received three reviews, the questionnaire for x solicited the review specific ratings (Part I of the questionnaire) three times – one time for each review. Authors had ten days to complete the questionnaire. We sent out one reminder eight days after the invitation.

As an incentive to participate in the study, besides that of helping the scientific community, we raffled off six Amazon gift certificates of USD 20,– each among all survey participants. We had announced the raffle in the invitation to the survey.

5.2.4. Statistical Methods

To quantify the pairwise relationships between variables, such as the characteristics of the reviews, the ratings, the accept/reject decision, etc., we perform a correlation analysis. Because most of the variables are ordinal in nature, we use Spearman's rank correlation coefficient ρ to calculate correlations between two variables [HWC13]. In line with the common practice, we refer to effects that have a significance level of $p \leq 0.05$ as *statistically significant*. Note that statistical significance does *not* refer to the size of the effect in question or its practical relevance. E.g., a weak correlation can still be statistically significant. In some situations we are interested in removing the effect of a third, confounding, variable on the correlation between two variables. To control for the effects of the third variable, we use partial correlation. We use Pearson's χ^2 test with Yates' continuity correction to compare differences in ratings and response rates between accepted and rejected submissions [FLP13].

5.3. Results

In the following, we present the results of our statistical analyses. To begin with, we present the response rate and an overview of the author ratings dealing directly with review satisfaction. Afterwards, we analyze the effects of review characteristics on ratings. Finally, we examine whether author estimates on rating distributions are in line with the common prior assumption.

5.3.1. Response Rate

We invited 72 authors of distinct papers to participate in the survey. In all, 39 out of the 72 invited authors completed the questionnaire, resulting in an overall response rate of 0.54. Authors of accepted papers were 2.2 times more likely to complete the questionnaire than authors of rejected papers (odds ratio 8.46, $\chi^2(1) = 11.95$, $p < 0.001$). Nevertheless, 46% of the respondents were authors of rejected papers. [G$^+$08] reports similar response rates. Overall, the authors assessed 175 reviews. The average number of assessed reviews per participating author was 4.49.

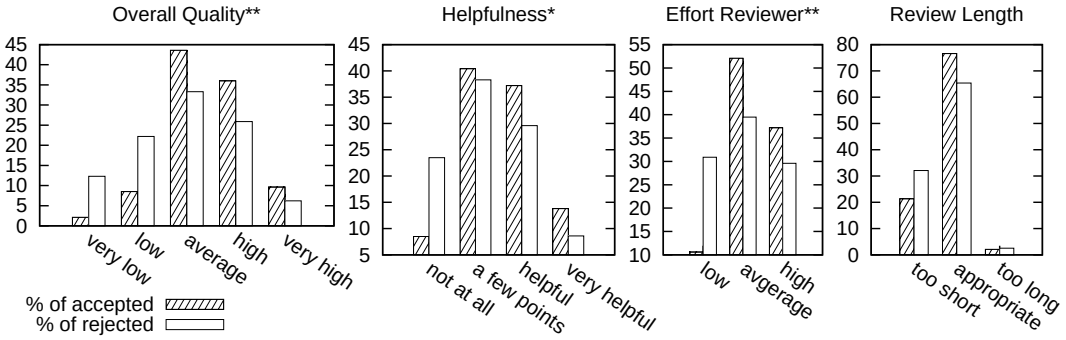


Figure 5.1.: Distributions of author ratings for review quality. Differences between ratings for accepted (filled) and rejected submissions are statistically significant ($^{*}p < 0.05$, $^{**}p < 0.01$) except for ratings of review length.

Rating	Correlation with *Overall Score*
Overall Quality	0.591**
Justified Comments	0.506**
Helpfulness	0.360**
Expertise Reviewer	0.417**
Effort Reviewer	0.482**

Table 5.2.: Correlations of quality ratings with the *Overall Score*. ($^{**}p < 0.01$)

5.3.2. Distribution of Review Satisfaction among Authors

Figure 5.1 shows an overview of the distributions of the four author ratings that are related to review quality, categorized by accepted and rejected submissions. The mean value of the fifth rating related to review quality, *Percentage of Justified Comments*, is 63.67 (standard deviation 17.67). Authors find 39% of reviews to be of high or very high quality and deem 45% of review comments helpful or very helpful. Further, they think that 34% of the reviews show that reviewers made a high effort to understand their paper. Finally, they deem 71% of the review comments to be of appropriate length. These findings suggest that authors are quite satisfied with the quality of the reviews their submissions received. Nevertheless, there seems to be room for improvement: authors rate 22% of reviews to be of low or very low quality, and 15% of the reviews as being not helpful at all.

5.3.3. Influence of the *Overall Score* on Quality Ratings

Table 5.2 shows the dependency of the ratings related to review quality on the *Overall Score*. All ratings show a statistically significant positive correlation with the *Overall Score*. In other words, unsurprisingly, authors tend to assign higher ratings to reviews

that assign high scores. But the correlations are not perfect and vary between rating categories. The *Overall Quality* rating shows the highest correlation, helpfulness has the lowest one.

5.3.4. Which Ratings Do Explain the *Overall Quality*?

The *Overall Quality* rating reflects the overall quality of the review as perceived by the author. By computing the correlation between the *Overall Quality* rating and each of the other author ratings, we can determine the criteria with the highest influence on the review quality as perceived by the author. However, as Section 5.3.3 shows, the author ratings for review quality depend on the *Overall Score* assigned by the reviewer. To remove this effect, we also computed the partial correlations between *Overall Quality* and each of the other author ratings, while controlling for the *Overall Score*.

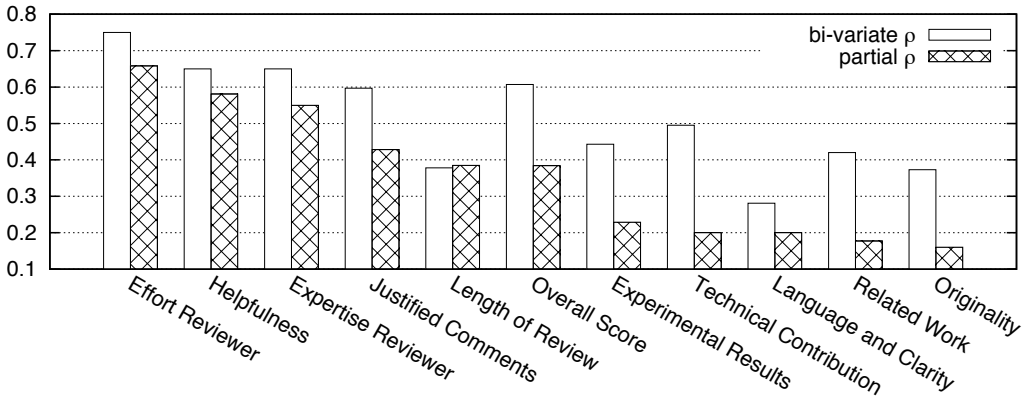


Figure 5.2.: Correlation of ratings with *Overall Quality* – bi-variate and controlled for *Overall Score*.

Figure 5.2 shows the results of both the bi-variate and the partial correlations. The light bars show the correlation between the respective rating of the authors and their *Overall Quality* rating. The filled bars show the same correlation when controlled for the effect of the *Overall Score*. The difference between the respective bars shows how big this effect is. For instance, the difference for the helpfulness rating is relatively small. This means that the correlation of *Helpfulness* with *Overall Quality* is rather independent of the *Overall Score*. In contrast, the *Overall Score* strongly influences the correlation of the rating for *Technical Contribution* with *Overall Quality*. All correlations between *Overall Quality* and the other author ratings are statistically significant. Ratings for *Effort of Reviewer*, *Helpfulness*, and *Expertise of Reviewer* show the highest correlation with the perceived review quality – both in the bi-variate case and when controlled for *Overall Score*.

5.3.5. Influence of Acceptance Status on Author Ratings

As Figure 5.1 shows, authors of rejected submissions assign lower quality ratings than those of accepted ones. This effect is statistically significant.

Further, we quantify the dependence of all rating categories on the accept/reject decision per correlation analysis. The acceptance status refers to the submission (as opposed to the review scores and author ratings that refer to individual reviews). Therefore, we computed averages of the review-specific author ratings per submission to test for correlation with the acceptance status.

We find that the correlations of acceptance status with the respective mean ratings per submission are rather low. They range from 0.06 to 0.253. In particular, the correlations of acceptance with the mean values of *Overall Quality* and *Effort of Reviewer* are $\rho = 0.236$ and $\rho = 0.182$, respectively. Thus, the effect of accept/reject decisions on author ratings is weaker than the effect of the *Overall Score* (cf. Section 5.3.3).

This finding was unexpected to us to some degree, as we had anticipated a stronger effect. However, in retrospect it might be explainable by the following facts. In our study, authors rated individual reviews. Thus, they could differentiate between reviews that assigned scores in their favor and those that did not. Since reviews per submission vary in their scores, and authors apparently take this into account, acceptance has a weaker effect on ratings than the scores.

5.3.6. Influence of Review Length

Minimum, maximum, and mean length of review comments in characters were 0, 11604, and 1488 respectively (standard deviation=1213, median=1258). Review comments are very rarely perceived as too long. But in over one fourth of the cases, authors perceive them as too short (see Figure 5.1). The length of a review comment is positively correlated with the rating for review length ($\rho = 0.501$, $p < 0.01$). The partial correlation controlled for *Overall Score* is slightly less ($\rho = 0.433$). Thus, authors appear to prefer longer reviews.

5.3.7. Expertise of Reviewer – Self-Assessed vs. Perceived

Authors rated the reviewer expertise on the same scale as the reviewer. The self-assessment by the reviewer and the assessment by the author are moderately correlated ($\rho = 0.360$, $p < 0.001$). This is the *only* non-negligible correlation of the *self-assessed expertise* with all other variables we analyzed. In particular, we do not find any correlation of self-assessed expertise with ratings for *Review Quality*, *Helpfulness*, and *Justified Comments*.

On the other hand, the *perceived expertise*, as measured by the authors' ratings, is significantly ($p < 0.01$) partially correlated (controlled for the *Overall Score*) with *Effort of Reviewer* ($\rho = 0.571$), *Helpfulness* ($\rho = 0.523$), and *Justified Comments* ($\rho = 0.434$). Like other ratings, *Expertise of Reviewer* moderately depends on the *Overall Score* ($\rho = 0.417$, $p < 0.001$). In other words, authors deem reviewers that give them higher scores more competent.

Score	Correlation with Respective Rating
Overall Quality	0.612**
Originality	0.596**
Technical Contribution	0.672**
Experimental Results	0.620**
Related Work	0.568**
Language and Clarity	0.655**

Table 5.3.: Correlation of review scores with their respective ratings. (** $p < 0.01$)

5.3.8. Rating of Review Scores

Interestingly, given the choices 'too low', 'adequate', and 'too high', authors rate the six scores their submissions had received per review mostly as adequate. The number of ratings per score with value 'adequate' ranges from 66% to 77% for the respective scores.

Authors almost never perceive the scores they received as too high. Out of 175 ratings for *Overall Quality*, 4 had the value 'too high'. All 4 were assigned by different authors. Further, 18 of the 875 ratings on the five detail scores had the value 'too high', 8 of which were assigned in category *Language and Clarity*.

Review scores are significantly positively correlated with their respective ratings (see Table 5.3). This means that authors tend to rate high scores as adequate and low scores as too low. But considering that authors have rated scores directly, the correlations are lower than we had expected.

5.3.9. Authors' Estimations of Rating Ratios

To test whether authors' ratings are in line with Bayesian updating (see 'Empirical evidence for Bayesian updating' on page 18), we asked authors to estimate the ratios of reviews rated unfavorably, i.e., as 'very low' or 'low', for three different subsets of authors. That is, we asked authors the following three questions: "What is your estimate of the percentage of reviews rated 'low' or 'very low' in this survey for" (i) all papers, (ii) for papers that have been accepted, and (iii) for papers that have been rejected.

Authors' mean estimates for the three ratios are higher than the mean values observed (Table 5.4). In other words, authors think that their peers rate more unfavorably than they actually do. However, the overall tendency of the estimated ratios is the same as in the ratios observed, i.e., accepted submissions yield less unfavorable ratings on average than all ratings combined, and all ratings combined yield less unfavorable ratings on average than rejected submissions.

More importantly, regarding Bayesian updating, there is a statistically significant effect of an author's own *Overall Quality* ratings on his estimations regarding the *Overall Quality* ratings issued by other authors. We obtained this result by calculating the share of unfavorable *Overall Quality* ratings issued by an author and comparing it to his estimates. The respective correlations of this share with the three estimates are significant

	Observed	Estimated	
	Mean	Mean	Std. Deviation
Accepted	0.106	0.238	0.165
Rejected	0.346	0.474	0.174
All	0.217	0.354	0.176

Table 5.4.: Observed and estimated values of unfavorable ratings for *Overall Quality*.

and range from 0.374 to 0.422 ($p < 0.05$ for all). Put simply, the more unfavorable ratings an author issues, the more he expects others to do the same. This suggests that authors do indeed behave like Bayesian learners who use their own opinion to update a (common) prior.

5.4. Discussion

The analysis confirms our expectations regarding this variant of peer production to a large extent. Authors' assessments of reviews are biased. They depend on review scores – on the overall scores as well as on the detail scores – but only weakly on the acceptance status. We think that this is because the granularity of assessment was the review, not the submission. That is, authors are not so much affected by a rejection per se, but they differentiate between reviews in their favor and reviews not in their favor.

The correlations of the authors' ratings with the reviewers' scores are relatively moderate. We had expected them to be stronger. In so far, authors appear to be 'decently honest'. Some author ratings are relatively 'neutral' towards the review scores. These ratings are *Effort of Reviewer*, *Percentage of Justified Comments*, and the *Helpfulness* of the comments. They are relatively weakly influenced by the *Overall Score* of the review, compared to the other ratings. Moreover, their respective correlations with *Overall Quality* hold when controlled for the *Overall Score*.

We are surprised to find that the reviewer's self-assessed expertise is not correlated with any of the ratings except for one: the assessment of the reviewer's expertise by the author. Therefore, we speculate that revealing the reviewers' self-assessment to authors affects the opinion of the authors. This is akin to the 'seeing is believing'-effect discussed in [CLA+03]. To examine this issue further, future experiments could divide authors into two groups, and display the self-assessed expertise level of reviewers to one group only. Comparing the results of both groups would yield insights as to whether authors are indeed affected by the display of the self-assessment.

Next, in our study, we provided the authors with three choices to assess review scores: 'too low', 'adequate', and 'too high'. We did this mainly to find out how many authors would choose 'too high'. As we have learned from our study, these choices appear to be rather inadequate for a real rating system: the number of ratings being 'too high' is negligible, resulting effectively in a boolean rating scale.

Objective criteria to identify and remunerate high-quality reviews are difficult to find.

In the end, quality and helpfulness can only be perceived and assessed by authors. Other parties that are assumed to be objective in their assessment – such as journal editors – are rather unsuitable [CKG02] to increase the quality. On the other hand, authors are not objective either. Their assessments are influenced by review comments and scores. For example, with the exception of *Language and Clarity*, ratings for scores are relatively strongly affected by their respective score, as well as by the *Overall Score*. How much of this influence is due to the scores and how much is due to the written comments is hard to determine. To examine this, a future experiment would have to introduce experimental groups of authors who only see review comments and do not see the scores and vice versa.

However, in a real-world review process, it is very difficult to split the authors into two groups which are then treated differently. One could, however, try to eliminate the influence of the scores on the quality ratings. How this could be achieved is the topic of the next section.

5.5. Remuneration for Reviews

One important objective of ours behind this study was to identify criteria that might be suitable to reward high-quality reviews. The main question in this context is: how to decouple incentives to write high-quality reviews from incentives to give accurate scores?

We have shown that there is a positive correlation between reviewers' scores and authors' ratings of reviews. That is, authors like reviews that like their submissions. Thus, if one simply remunerated reviews based on how highly authors rate them, it would create incentives for reviewers to give inaccurately high scores. Consequently, we propose to remunerate *relatively* highly rated reviews, i.e., reviews that receive high ratings by authors despite assigning low scores. In the following, we formalize one possible function that achieves this. We explicitly write down this function for illustration purposes, and to indicate a potential direction of future research.

Let $r \in \{1, \ldots, k\}$ denote the value of the author rating of a given review. Let $s \in \{1, \ldots, l\}$ denote that review's score. The remuneration function $t(r, s) = r - s$ removes the influence of the score on the remuneration.

The function t can be further refined. For example, one could normalize the rating scales if $k \neq l$. Further, reviewers might be deterred from reviewing if threatened by penalties. So one could only remunerate good reviews, but refrain from any penalization. Alternatively, one could scale t such that all values $t(r, s)$ are non-negative.

In order to see whether our proposed remuneration indeed neutralizes the effects of scores, we apply t to the data of our study. Let r be the *Overall Quality* rating by an author and s the *Overall Score* of the respective review. Further, according to the number of different choices for ratings and scores in our study, we fix $k, l = 5$. Table 5.5 shows the results. The remuneration is quite symmetrically distributed. About 44.6% of the reviews would not be remunerated at all. Moreover, $t(r, s)$ is positively correlated with *Overall Quality* ($\rho = 0.526$) and weakly negatively correlated with *Overall Score* ($\rho = -0.333$).

A further decoupling of the incentives from scores could be achieved by choosing a

t	Reviews receiving t
-2	1.1%
-1	24.0%
0	44.6%
1	24.6%
2	5.7%

Table 5.5.: Function t applied to the data of our study. (Values less than -2 and greater than 2 did not occur.)

rating category for r that is only weakly dependent on s. One candidate, for example, is the *Helpfulness* of the comments for future work, because, of all author ratings, its dependency on the *Overall Score* is the weakest one. To demonstrate this, we use a normalized variant of the remuneration function above and apply it to the data of our study: Let $r_{\text{help}} \in \{1, \ldots, 4\}$ be the rating for the helpfulness of a given review, and let s be that review's *Overall Score*. The resulting remuneration for helpfulness

$$t_{\text{help}}(r_{\text{help}}, s) = \frac{r_{\text{help}}}{4} - \frac{s}{5}$$

is only negligibly dependent on the *Overall Score* ($\rho = -0.126$, $p = 0.096$), while still being strongly correlated with *Helpfulness* ($\rho = 0.761$, $p < 0.01$) and *Overall Quality* ($\rho = 0.591$, $p < 0.01$). Thus, it decouples the incentive to give accurate scores from the incentive to write high-quality reviews to a large degree. Of course, we formulated the functions t and t_{help} based on the data of our study. To ensure their validity, they need to be tested on data that are independent of the study at hand.

Two potential problems might arise given a remuneration function such as the one above:

(1) In our study, the perceived *Helpfulness* of review comments was weakly correlated with *Overall Score*. That is, authors appreciated helpful comments even if the reviewer gave them a low score. Now, if t_{help} is employed, authors might resort to assigning inaccurately low ratings for *Helpfulness* in order to punish reviewers for low scores. In that case, the weak association between *Helpfulness* and *Overall Score* would turn into a strong one. However, we deem such behavior of authors unlikely since it would be irrational: As long as authors do not pay the reviewers' remuneration themselves, they have no incentive to resort to assigning dishonestly low ratings. Moreover, they cannot incentivize reviewers to assign a higher *Overall Score* by threatening them with low *Helpfulness* ratings, since t_{help} specifically removes the effects of high scores from the remuneration.

(2) According to t and t_{help}, all else being equal, reviewers could increase their chance of being remunerated by assigning lower scores. In the worst case, all reviewers would assign minimum scores while still trying to write helpful comments. Clearly, this is undesirable. To counter artificially low scores, conferences could use an HRM (cf. Section 2.2). In this case, some of the remuneration for a review would be based on its score in comparison to

the scores of other reviews for the same submission. Reviewers would then face a trade-off between two factors: Some of the remuneration would be based on author ratings, some based on review scores. Studying the question how this trade-off influences reviewer behavior is beyond the scope of this chapter: The specifics of the remuneration function, in particular the proposal how it might depend on review scores, are a result of our study. We had not foreseen them prior to the study and hence had not incorporated them in the questionnaire. Thus, the reviewers in our study did not face the trade-off described above. Consequently, we could not study the trade-off based on the data we collected about them.

Finally, for future work, we deem experiments the most promising way to study reviewer behavior in presence of the trade-off described above. I.e., we would let reviewers know the remuneration function(s) and measure how this affects their behavior.

5.6. Conclusions

Selecting conference articles is an important instance of peer production. Today, this is typically done by means of peer reviewing. Review ratings by authors have potential to improve the quality of peer reviews. A significant problem, however, is that authors' perception is hardly neutral, but might in turn be affected by the reviews. To gain empirical insight into authors' perception of reviews, we have conducted a study with 39 authors of a computer science conference who rated 175 reviews they had received. The results of this study show that the authors' satisfaction with review quality is good, but leaves some room for improvement. Review scores affect author ratings to different degrees. Authors rate reviews as good if they deem the review helpful for their future work, if they deem the review comments justified, and if they have the impression that the reviewer made an effort to understand the paper. By and large, these results hold when controlled for the overall score. Acceptance and self-assessed reviewer expertise only have a weak influence on perceived review quality. Finally, we find that the assumption that authors act in line with Bayesian updating, which is crucial for HRMs, holds with respect to authors. Given these results of the study, we have discussed suitable metrics to compute remunerations for reviews based on ratings and scores. As discussed, such metrics are not obvious without studying author behavior first. Applied to the data collected in our study, one of these metrics neutralizes the effects of scores to a large degree. A possible limitation of our study is that we only looked at one conference. The findings might differ for other conferences. Even though, we belief that it is reasonable to assume that results for other conferences would point into the same direction. Future work would have to verify this.

6. Conclusion

In this dissertation, we have investigated the problems of user motivation and quality assurance in three related peer-production settings.

To meet the problems of user motivation and contribution quality, we have proposed the usage of ratings and rating-based incentive mechanisms for the creation of structured knowledge. To evaluate our approach, we have developed a platform and conducted a study consisting of six different controlled field experiments. The results of the experimental study show that our platform is well-suited for the collaborative creation of structured knowledge. We find that the usage of rating mechanisms, as well as fully rating-dependent rewards for good contributions, increase the quality of contributions. Further, we find that an honest rating mechanism improves the quality of ratings in most of the experiments. Thus, our results provide a contribution to the fundamental research issue of how to create high-quality structured knowledge on a large scale.

We have investigated the problem of classifying contributions in an open peer-rating online community. In particular, we have studied how to increase the classification accuracy of the Dawid-Skene algorithm in the presence of low-competence raters, such as spammers or biased raters. We have proposed and evaluated gold strategies based on the level of agreement to increase the accuracy of the Dawid-Skene algorithm. Further, in order to maximize the net benefit of gold objects, i.e., their benefit minus their costs, we have proposed and evaluated an adaptive algorithm. It determines the number of gold objects based on runtime information Our main finding is that the HI-ALO gold strategy is very effective in increasing the accuracy of the Dawid-Skene algorithm in low-competence settings. Further, we find that the HI-ARS and the UNI strategy are effective in reducing the benefit gained by colluders, thus, rendering collusions less desirable. Moreover, the adaptive algorithm determines the optimal gold ratio for each strategy and each setting with high accuracy.

We have conducted an empirical study on how authors rate the peer reviews they have received, and how the author's ratings, in turn, can be used to increase the quality of the reviews. To this end, we have incorporated review ratings into the review process of a computer science conference. We find that review scores affect author ratings to different degrees. Further, we find that authors rate reviews as good if they deem them helpful for their future work, or if they have the impression that the reviewer made an effort to understand the paper. By and large, the latter results hold when controlled for the overall review score. Surprisingly, acceptance and self-assessed reviewer expertise only have a weak influence on perceived review quality. Given these results of the study, we have discussed suitable metrics to compute remunerations for reviews based on ratings and scores. Applied to the data collected in our study, on of the metrics neutralizes the effects of scores to a large degree. In the future, remunerations based on this metric

could increase the quality of the peer-review process.

In summary, this dissertation has shown how ratings and rating-based incentive mechanisms can increase the quality and the quantity of contributions in peer-production settings.

Appendix A.

Further Figures of Consensus Builder

Figures A.1 to A.3 show further screenshots of Consensus Builder.

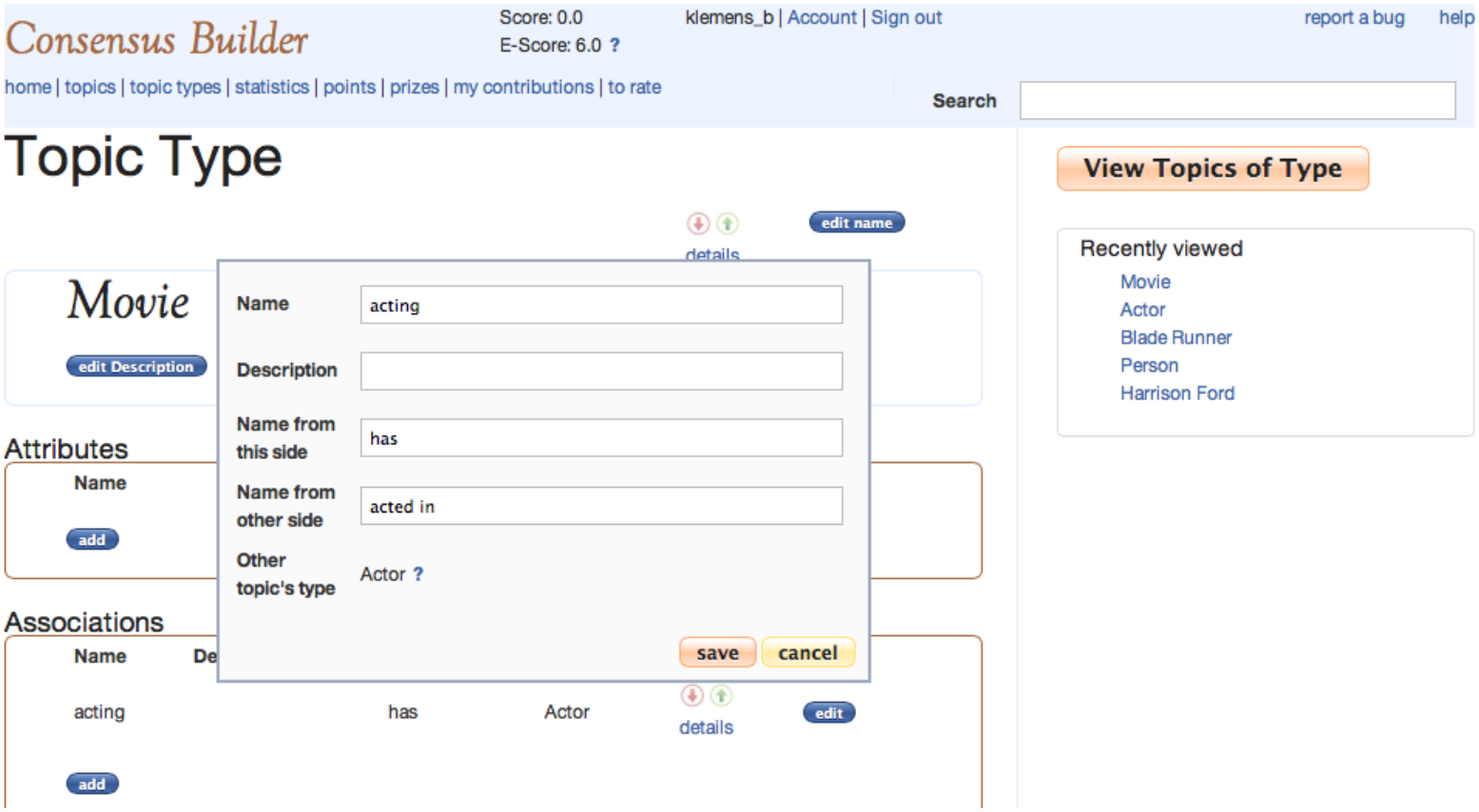


Figure A.1.: Screenshot Consensus Builder 2.0: Editing association `acting` of topic type `Movie`.

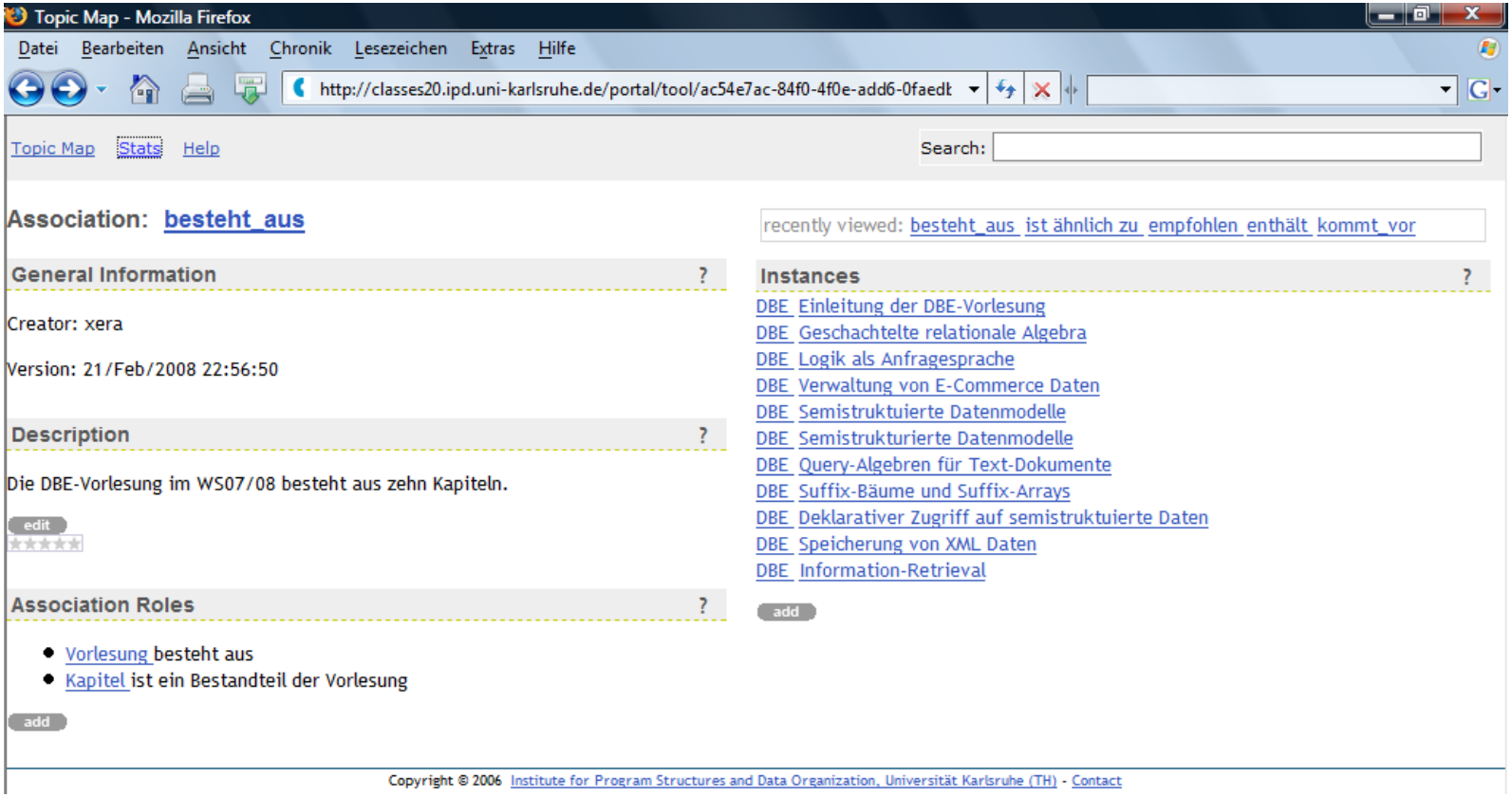


Figure A.2.: Screenshot Consensus Builder 1.0: Details for association `besteht aus`.

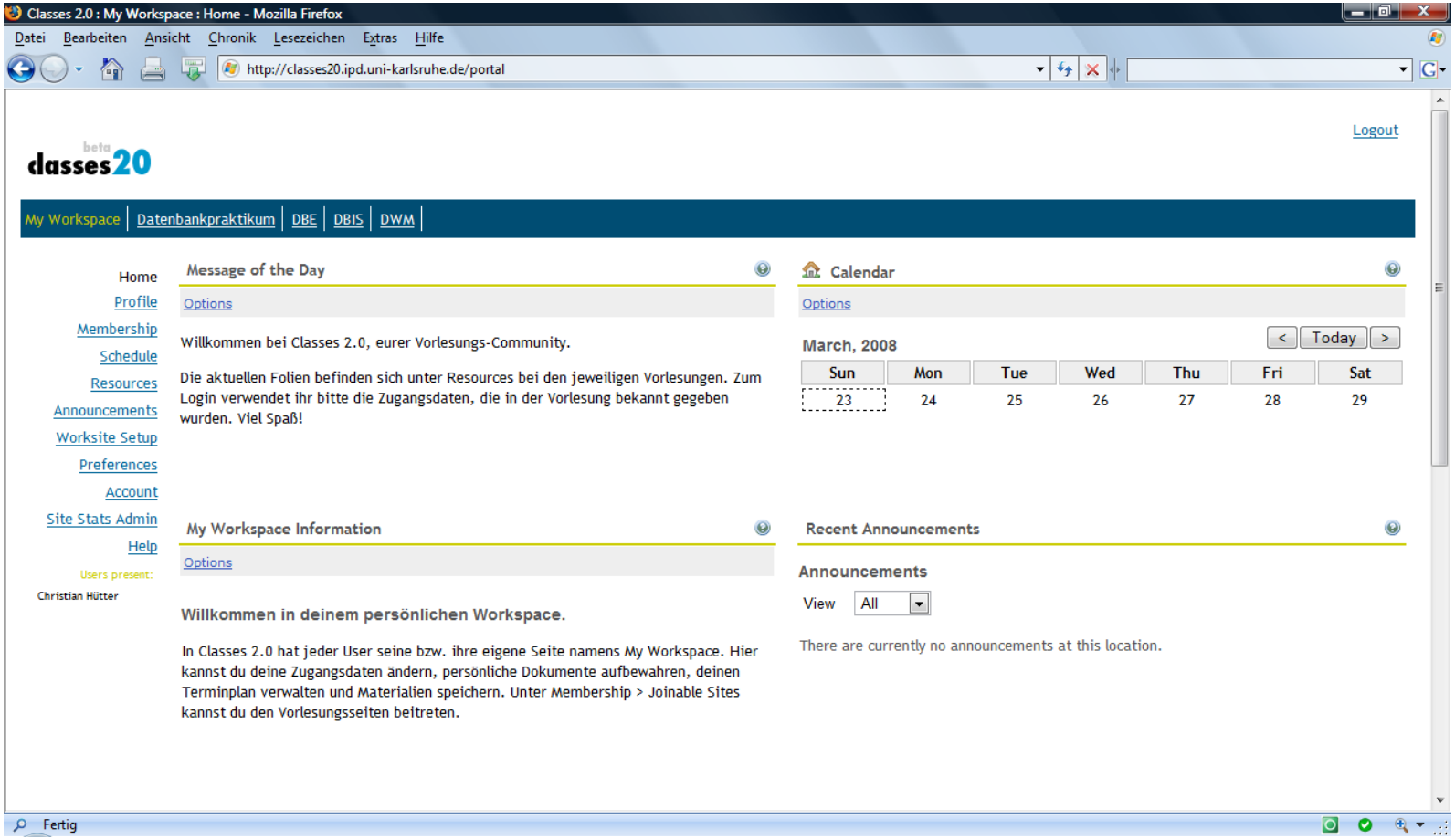


Figure A.3.: Lecture community portal ‘classes 2.0’.

Appendix B.

Proofs

B.1. Proof of Proposition 1

Proof. For simplicity and brevity, we omit the subscript k in the following. Thus, s_k becomes s, R_k becomes R, o_k becomes o, $r_{i,k}$ becomes r_i, and so on. Further, let i refer to the s raters of the data object in question.

The posterior probability that the object is of type t given the ratings is

$$P(o = t \mid R) = \frac{P(R \mid o = t)P(o = t)}{P(R)}.$$

Since $P(R)$ is the same for both types, the ratio of the posterior probabilities (or posterior probability ratio, *ppr*) of the data object being of type -1 and being of type 1 given its ratings is

$$ppr = \frac{P(o = -1 \mid R)}{P(o = 1 \mid R)} = \frac{P(o = -1)}{P(o = 1)} \frac{P(R \mid o = -1)}{P(R \mid o = 1)}.$$

Since we assume conditional independence of the ratings given a type, their joint probability can be written as a product

$$ppr = \frac{P(o = -1)}{P(o = 1)} \prod_{i=1}^{s} \frac{P(r_i \mid o = -1)}{P(r_i \mid o = 1)}. \tag{B.1}$$

Per definition of the homogeneous competence we have $P(r_i = q \mid o = t) = c$, if $q = t$, and $1 - c$ otherwise. Thus, we can express each *likelihood ratio* as

$$\frac{P(r_i = q \mid o = -1)}{P(r_i = q \mid o = 1)} = \begin{cases} c/(1-c) & \text{if } q = -1 \\ (1-c)/c & \text{if } q = 1. \end{cases}$$

Let $lrr = \prod_{i=1}^{s} P(r_i \mid o = -1)/P(r_i \mid o = 1)$ denote the product of the likelihood ratios of all s ratings. Since s is odd and $c > 0.5$ and therefore $c/(1-c) > 1$ the likehood ratio llr cannot be equal to 1. Instead either

- $lrr \geq (c)/(1-c) > 1$, if more ratings are in favor for type -1, or
- $lrr \leq (1-c)/c < 1$, if more ratings are in favor for type 1.

The same is true for the *ppr*. This is because we assume that $1 - c < P(o = t) < c$ for both types $t \in \{-1, 1\}$. Thus, the ratio of the priors is $(1 - c)/c < P(o = -1)/P(o = 1) < c/(1 - c)$ and therefore cannot "reverse the direction" of the *ppr*: If $lrr > 1$, then $ppr = lrr \cdot P(o = -1)/P(o = 1) > 1$. Likewise, if $lrr < 1$, then $ppr = lrr \cdot P(o = -1)/P(o = 1) < 1$.

Thus, the posterior probability is greater for the type that receives the majority of ratings in its favor. □

B.2. Proof of Proposition 2

Proof. For simplicity and brevity, we omit the subscript k in the following. Thus, s_k becomes s, l_k becomes l, ars_k becomes ars, and so on.

Let $corr(\hat{o} = o)$ denote the number of correct ratings for a correct classification by MV. Correspondingly, let $corr(\hat{o} \neq o)$ denote the number of correct ratings for an incorrect classification by MV. We first proof the following lemma.

Lemma 1. *For a given ars and a given odd s, if the number of correct ratings for an incorrect classification by MV is $corr(\hat{o} \neq o) = l$, then the number of correct ratings for a correct classification by MV is $corr(\hat{o} = o) = s - l$. Further, $l = (s - ars)/2$.*

Proof. Let rs denote the rating sum for a given ars, i.e.,

$$ars = |rs| = \begin{cases} -rs & \text{if } rs < 0 \\ rs & \text{if } rs > 0 \end{cases}. \tag{B.2}$$

Note that since the number of ratings is odd, rs cannot be 0. Let rs^+ denote positive rating sum, i.e., the rating sum $rs > 0$ in Equation (B.2). Further, let s^- be the number of negative ratings of rs^+, that is, the number of ratings that equal -1 if $rs > 0$. Thus, the number of negative ratings for rs^+ is $(s - s^-)$. The positive rating sum is the sum of the negative ratings and the positive ratings

$$rs^+ = -s^- + (s - s^-) \tag{B.3}$$

with the number of negative ratings of rs^+ being the smaller of the two summands $0 \leq s^- < s - s^-$.

Note, that there is at most one s^- for a given rs^+ and a given s. To see why this is true, suppose, for contradiction, there is a second $s^- := s^- + k$, for some integer $k \neq 0$. Then, the number of positive ratings for rs^+ is $(s - (s^- + k))$. But the sum of the number of negative ratings and the number of positive ratings must be $s = s^- + k + (s - (s^- + k))$ which can only be true if $k = 0$.

The negative rating sum rs^-, i.e., the $rs < 0$, is

$$\begin{aligned} rs^- &= -rs^+ \\ &= -(s - s^-) + s^-. \end{aligned} \tag{B.4}$$

Hence, s^- is the number of positive ratings of rs^- and $(s - s^-)$ is the number of negative ratings of rs^-.

For a given ars and a given s, let $corr(\hat{o} \neq o) = l$ be the number of correct ratings for an incorrect classification. For incorrect classification by MV, the number of correct ratings must be less than the number of incorrect ratings. Thus, l must be the smaller one of the two terms s^- and $(s - s^-)$, i.e., $l = s^-$. For the correct classification by MV, the number of correct ratings must be the larger of the two terms, i.e., $corr(\hat{o} = o) = s - s^- = s - l$.

To prove the second part of Lemma 1, we use the relationship between ars and rs^- and rs^+ defined in Equation (B.2). We substitute $s^+ = s - l$ and $s - s^+ = l$ in Equation (B.3)

$$\begin{aligned} ars = rs^+ &= -l + (s - l) \\ &= s - 2l \end{aligned}$$

and in Equation (B.4)

$$\begin{aligned} ars = -rs^- &= -(-(s - l) + l) \\ &= s - 2l. \end{aligned}$$

Rearranging, either one of the above equations proves the second part of the lemma: $l = (s - ars)/2$. □

The probability of an incorrect classification by MV given ars is

$$P(\hat{o} \neq o \mid ars) = \frac{P(\hat{o} \neq o, ars)}{P(ars)} = \frac{P(\hat{o} \neq o, ars)}{P(\hat{o} \neq o, ars) + P(\hat{o} = o, ars)}. \tag{B.5}$$

As Lemma 1 shows, to obtain a given ars out of s ratings, we need exactly $l = (s - ars)/2$ correct ratings. The probability that exactly l out of s ratings are correct for a given competence c, is $\binom{s}{l} c^l (1 - c)^{(s-l)}$ (see Equation (4.2)). Thus, the joint probability of MV classifying the data object incorrectly and having rating sum ars is

$$P(\hat{o} \neq o, ars) = \binom{s}{l} c^l (1 - c)^{s-l}.$$

where $l = (s - ars)/2$. By Lemma 1, the joint probability of MV classifying the data object correctly and having rating sum ars is

$$P(\hat{o} = o, ars) = \binom{s}{s - l} c^{s-l} (1 - c)^l.$$

Substituting the two equations above in Equation (B.5), we obtain

$$P(\hat{o} \neq o \mid ars) = \frac{\binom{s}{l} c^l (1 - c)^{s-l}}{\binom{s}{l} c^l (1 - c)^{s-l} + \binom{s}{s-l} c^{s-l} (1 - c)^l}.$$

$$P(\hat{o} \neq o \mid ars) = \frac{c^l (1 - c)^{s-l}}{c^l (1 - c)^{s-l} + c^{s-l} (1 - c)^l},$$

where $l = (s - ars)/2$. □

Appendix C.

Summary of Notation of Chapter 4

Table C.1 summarizes the most frequently used symbols of this chapter and their meanings.

Symbol	Meaning
$T = \{-1, 1\}$	Binary set of types
$q, t \in T$	Type q and type t
$p(t)$	Prior probability of type t
$K = \{1, \ldots, m\}$	Set of data objects
$k \in K$	Data object
o_k	True type of k
$m = \lvert K \rvert$	Number of data objects
I	Set of raters $I = \{1, \ldots, n\}$
i	Rater i
$n = \lvert I \rvert$	Number of raters
$r_{i,k}$	Rating that rater i gives to data object k
$R = \{r_{i,k}\}$	Set of all ratings
$R_k = \{r_{i,k'} \mid k' = k\}$	The set of ratings of data object k
$s_k = \lvert R_k \rvert$	Number of ratings given to data object k
l_k	Number of correct ratings for k
$c_i^{(t)}$	Type dependent competence, i.e., probability $P(r_{i,k} = t \mid o_k = t)$ that rater i rates objects of type t correctly
c_i	Type-independent competence, i.e., probability that rater i rates correctly
c	Homogeneous competence, i.e., competence $c = c_i$ that is the same for all raters i
g	Gold ratio, i.e., ratio of gold objects
K_{gold}	Set of gold objects
$m_{\text{gold}} = \lvert K_{\text{gold}} \rvert$	Number of gold objects
$I_{\text{col}} \subseteq I$	The subset of colluders among all raters
$K_{\text{col}} \subseteq K$	The subset of data objects that the colluders use for the collusion attack
$K_{\text{hon}} = K \setminus K_{\text{col}}$	The set of data objects rated by honest raters only
$n_{\text{col}} = \lvert I_{\text{col}} \rvert$	Number of colluders
$m_{\text{col}} = \lvert K_{\text{col}} \rvert$	Number of collusion objects
m_{col}/m	Ratio of collusion objects
$\hat{\theta}$	An estimator of a given parameter θ
$\mathbb{1}(\cdot)$	Indicator function, i.e., $\mathbb{1}(\cdot)$ is equal to one if its argument holds true, and equal to zero otherwise

Table C.1.: Symbols and meanings.

Appendix D.

CASES 2009 Review Survey

The following is an offline version of our CASES 2009 survey.

Welcome to our survey of the CASES 2009 reviews!

In the following we ask nine questions for each review having to do with review quality. The numbering of the reviews in this survey refers to the order of the reviews in your notification email/in the invitation email for this survey. Of course, we would like to hear your honest opinion.

Questions marked with a * are mandatory.

Review 1

What is your overall rating of the quality of Review 1?*

Please choose **only one** of the following:

- very low
- low
- average
- high
- very high

How do you assess the overall score that Review 1 has given to your paper?*

Please choose **only one** of the following:

- too low
- adequate
- too high

How do you assess the detail ratings that Review 1 has given to your paper?*

Please choose the appropriate response for each item:

	too low	adequate	too high
Originality	○	○	○
Technical Contribution	○	○	○
Experimental Results	○	○	○
Description Related Work	○	○	○
Language and Clarity	○	○	○

What is the percentage of comments in Review 1 which you find justified?*

Please choose **only one** of the following:

- 0%
- 25%
- 50%
- 75%
- 100%

Do you find Review 1 helpful for your future work?*

Please choose **only one** of the following:

- not at all helpful
- a few points are helpful
- helpful
- very helpful

What is your opinion about the level of expertise of the reviewer?*

Please choose **only one** of the following:

- 1 (Minimal)
- 2
- 3
- 4
- 5 (Maximal)

Do you think that the reviewer has made an effort to understand your paper?*

Please choose **only one** of the following:

- low effort
- average effort
- high effort

What do you think of the length of Review 1?*

Please choose **only one** of the following:

- too short
- appropriate
- too long

Are there any other reasons why you are particularly happy/particularly unsatisfied with this review?

Please write your answer here:

Review 2

What is your overall rating of the quality of Review 2?*

...

General Questions

Estimation Questions

The following three questions solicit your estimate of the relative frequencies of answers to the question "What is your overall rating of the quality of the review?" (which we have just asked) from all authors who take part in this survey.

What is your estimate of the percentage of reviews rated 'low' or 'very low' in this survey?*

Please choose **only one** of the following:

- 0%
- 10%

...

- 90%
- 100%

What is your estimate of the percentage of reviews rated 'low' or 'very low' in this survey for papers that have been accepted?*

Please choose **only one** of the following:

- 0%
- 10%

...

- 90%
- 100%

What is your estimate of the percentage of reviews rated 'low' or 'very low' in this survey for papers that have been rejected?*

Please choose **only one** of the following:

- 0%
- 10%

...

- 90%
- 100%

Do you deem it likely that rating reviews will improve review quality?*

Please choose **only one** of the following:

- Yes
- No

[Submit your survey]

Bibliography

[AD06] S. Auer and S. Dietzold, "OntoWiki - a tool for social, semantic collaboration," in *Proceedings of the 5th International Semantic Web Conference ISWC*, 2006.

[AH07] S. Auer and H. Herre, "RapidOWL – an agile knowledge engineering methodology," in *Perspectives of Systems Informatics*. Springer, 2007, pp. 424–430.

[AK97] M. Z. Al Kawi, "History of medical records and peer review," *Ann. Saudi. Med*, vol. 17, pp. 277–8, 1997.

[AMP06] I. B. Aban, M. M. Meerschaert, and A. K. Panorska, "Parameter estimation for the truncated Pareto distribution," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 270–277, 2006. [Online]. Available: http://amstat.tandfonline.com/doi/abs/10.1198/016214505000000411

[AP00] A. Admati and P. Pfleiderer, "Noisytalk. com: Broadcasting opinions in a noisy environment," 2000.

[ASM11] A. Albers, E. Sadowski, and L. Marxen, "A new perspective on product engineering overcoming sequential process models," in *The Future of Design Methodology*, H. Birkhofer, Ed. Springer London, 2011, pp. 199–209. [Online]. Available: http://dx.doi.org/10.1007/978-0-85729-615-3_17

[AVH04] G. Antoniou and F. Van Harmelen, *A Semantic Web Primer*, ser. Cooperative information systems. MIT press, 2004.

[B+09] C. Bauer *et al.*, "The student view on online peer reviews," in *ITiCSE'09*. Paris, France: ACM, 2009.

[BBC+07] D. J. Benos, E. Bashari, J. M. Chaves, A. Gaggar, N. Kapoor, M. LaFrance, R. Mans, D. Mayhew, S. McGowan, A. Polter *et al.*, "The ups and downs of peer review," *Advances in physiology education*, vol. 31, no. 2, pp. 145–152, 2007.

[Ben02] Y. Benkler, "Coase's penguin, or Linux and the nature of the firm," *Yale Law Journal*, pp. 369–446, 2002.

[Ben06] ——, *The wealth of networks: How social production transforms markets and freedom*. Yale University Press, 2006.

[BEP+08] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 1247–1250.

[BLF99] T. Berners-Lee and M. Fischetti, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, 1st ed. Harper San Francisco, 1999.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[BLW+04] G. Beenen, K. Ling, X. Wang, K. Chang, D. Frankowski, P. Resnick, and R. E. Kraut, "Using social psychology to motivate contributions to online communities," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04*. New York, New York, USA: ACM Press, 2004, p. 212. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1031607.1031642

[BM05] P. Buitelaar and B. Magnini, "Ontology learning from text: An overview," in *Ontology Learning from Text: Methods, Applications and Evaluation*. IOS Press, 2005, pp. 3–12.

[BMS07] J. Bhogal, A. Macfarlane, and P. Smith, "A review of ontology based query expansion," *Information processing & management*, vol. 43, no. 4, pp. 866–886, 2007.

[BPSW10] P. Brusilovsky, D. Parra, S. Sahebi, and C. Wongchokprasitti, "Collaborative information finding in smaller communities: The case of research talks," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*. IEEE, 2010, pp. 1–10.

[Bra11] S. Braun, "Community-driven & work-integrated creation, use and evolution of ontological knowledge structures," Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, Nov. 2011. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000025701

[BS91] H. Burkhardt and B. Smith, *Handbook of metaphysics and ontology*, ser. Analytica. Philosophia Verlag, 1991, no. v. 1. [Online]. Available: http://books.google.de/books?id=zCUOAQAAMAAJ

[BSW+07] S. Braun, A. Schmidt, A. Walter, G. Nagypal, and V. Zacharias, "Ontology maturing: a collaborative Web 2.0 approach to ontology engineering," in *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 07), Banff, Canada*, 2007.

[Cap13] V. Capraro, "A model of human cooperation in social dilemmas," *PloS one*, vol. 8, no. 8, p. e72427, 2013.

[CAS09] "CASES 2009 - international conference on compilers, architecture, and synthesis for embedded systems," 2009.

[CBP01] J. Cameron, K. M. Banko, and W. D. Pierce, "Pervasive negative effects of rewards on intrinsic motivation: The myth continues," *The Behavior Analyst*, vol. 24, no. 1, p. 1, 2001.

[CBWW98] M. L. Callaham, W. G. Baxt, J. F. Waeckerle, and R. L. Wears, "Reliability of editors' subjective quality ratings of peer reviews of manuscripts," *JAMA*, vol. 280, no. 3, 1998.

[CCT04] S. E. Campanini, P. Castagna, and R. Tazzoli, "Platypus Wiki: a Semantic Wiki Wiki Web," in *1st Italian Semantic Web Workshop*, 2004.

[CFK+05] D. Cosley, D. Frankowski, S. Kiesler, L. Terveen, and J. Riedl, "How oversight improves member-maintained communities," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '05. New York, NY, USA: ACM, 2005, pp. 11–20.

[Che76] P. P.-S. Chen, "The entity-relationship model—toward a unified view of data," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[CKG02] M. L. Callaham, R. K. Knopp, and E. J. Gallagher, "Effect of written feedback by editors on quality of reviews: Two randomized trials," *JAMA*, vol. 287, no. 21, 2002.

[CLA+03] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl, "Is seeing believing?: How recommender system interfaces affect users' opinions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 585–592. [Online]. Available: http://doi.acm.org/10.1145/642611.642713

[CMSV09] P. Cimiano, A. Mädche, S. Staab, and J. Völker, "Ontology learning," in *Handbook on ontologies*. Springer Berlin Heidelberg, 2009, pp. 245–267.

[Coh68] J. Cohen, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit." *Psychological bulletin*, vol. 70, no. 4, p. 213, 1968.

[Con85] J.-A.-N. d. C. M. d. Condorcet, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: Imprimerie royale, 1785. [Online]. Available: http://gallica.bnf.fr/ark:/12148/bpt6k417181

[Coo91] R. M. Cooke, *Experts in uncertainty: opinion and subjective probability in science*. New York, NY (United States); Oxford University Press, 1991.

[CP02] J. Cameron and W. Pierce, *Rewards and Intrinsic Motivation: Resolving the Controversy*. BERGIN & GARVEY, 2002.

[CS02] V. Conitzer and T. Sandholm, "Complexity of mechanism design," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 103–110.

[CSN09] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1137/070710111

[Daw89] R. M. Dawes, "Statistical criteria for establishing a truly false consensus effect." *J. Exp. Soc. Psychol*, no. 25, 1989.

[Daw90] ——, "The potential nonfalsity of the false consensus effect," *Insights in decision making: A tribute to Hillel J. Einhorn*, p. 179, 1990.

[Dec71] E. L. Deci, "Effects of externally mediated rewards on intrinsic motivation," *Journal of Personality and Social Psychology*, vol. 18, no. 1, pp. 105–115, 1971.

[DFH11] J. Domingue, D. Fensel, and J. A. Hendler, Eds., *Handbook of Semantic Web Technologies*. Berlin: Springer, 2011.

[DH73] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, 1st ed. John Wiley & Sons Inc, Feb. 1973.

[DHS07] C. V. Damme, M. Hepp, and K. Siorpaes, "Folksontology: An integrated approach for turning folksonomies into ontologies," in *Proceedings of the ESWC Workshop "Bridging the Gap between Semantic Web and Web 2.0"*. Springer, 2007.

[DKND11] S. Deterding, R. Khaled, L. E. Nacke, and D. Dixon, "Gamification: Toward a definition," *CHI 2011 Gamification Workshop Proceedings*, 2011.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[Dou02] J. R. Douceur, "The sybil attack," in *IPTPS*, 2002.

[DR85] E. L. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior*, ser. Perspectives in Social Psychology. Springer, 1985. [Online]. Available: http://books.google.de/books?id=p96Wmn-ER4QC

[DR00] ——, "The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior," *Psychological Inquiry*, vol. 11, no. 4, pp. 227–268, 2000.

[DS79] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pp. 20–28, 1979.

[dWD10] J. C. de Winter and D. Dodou, "Five-point Likert items: t Test versus Mann-Whitney-Wilcoxon," *Practical Assessment, Research & Evaluation*, vol. 15, no. 11, 2010.

[E+10] K. Eckert *et al.*, "Crowdsourcing the assembly of concept hierarchies," in *Proceedings of the 10th annual joint conference on Digital libraries - JCDL '10*, 2010.

[ECD+04] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Web-scale information extraction in KnowItAll (preliminary results)," in *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004, pp. 100–110.

[Efr71] B. Efron, "Forcing a sequential experiment to be balanced," *Biometrika*, vol. 58, no. 3, pp. 403–417, Dec. 1971.

[FDM+08] R. Farzan, J. M. DiMicco, D. R. Millen, C. Dugan, W. Geyer, and E. A. Brownholtz, "Results from deploying a participation incentive mechanism within the enterprise," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 563–572. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357145

[FDP+05] T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng, "Swoogle: Searching for knowledge on the Semantic Web," in *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, vol. 20, no. 4. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1682.

[Fel98] C. Fellbaum, *WordNet: An Electronic Lexical Database*, ser. Language, speech, and communication. MIT Press, 1998. [Online]. Available: http://books.google.de/books?id=Rehu8OOzMIMC

[Flo04] L. Floridi, *The Blackwell Guide to the Philosophy of Computing and Information*, ser. Blackwell Philosophy Guides. Wiley, 2004. [Online]. Available: http://books.google.de/books?id=rIbJJOjoqygC

[FLP13] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical methods for rates and proportions*. John Wiley & Sons, 2013.

[FT91] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, 1991.

[G+90] J. Garfunkel *et al.*, "Effect of acceptance or rejection on the author's evaluation of peer review of medical manuscripts," *JAMA*, vol. 263, no. 10, 1990.

[G+08] M. Gibson *et al.*, "Author perception of peer review." *Obstetrics and gynecology*, vol. 112, no. 3, Sep. 2008.

[Gal04] D. Galin, *Software Quality Assurance: From Theory to Implementation.* Harlow, UK: Pearson Education, 2004.

[GAVS11] S. Grimm, A. Abecker, J. Völker, and R. Studer, "Ontologies and the Semantic Web," in *Handbook of Semantic Web Technologies.* Springer, 2011, pp. 507–579.

[GG95] N. Guarino and P. Giaretta, "Ontologies and knowledge bases: Towards a terminological clarification," in *Towards very Large Knowledge bases: Knowledge Building and Knowledge sharing.* IOS Press, 1995, pp. 25–32.

[Gil13] E. Gilbert, "Widespread underprovision on reddit," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 803–808. [Online]. Available: http://doi.acm.org/10.1145/2441776.2441866

[GL02] M. Grüninger and J. Lee, "Ontology applications and design - introduction," *Commun. ACM*, vol. 45, no. 2, pp. 39–41, 2002.

[GN87] M. R. Genesereth and N. J. Nilsson, *Logical foundations of artificial intelligence.* Morgan Kaufmann Los Altos, CA, 1987, vol. 9.

[GOF83] B. Grofman, G. Owen, and S. L. Feld, "Thirteen theorems in search of the truth," *Theory and Decision*, vol. 15, no. 3, pp. 261–278, 1983. [Online]. Available: http://dx.doi.org/10.1007/BF00125672

[GOS09] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" in *Handbook on ontologies.* Springer, 2009, pp. 1–17.

[GP00] R. A. Ghosh and V. V. Prakash, "The orbiten free software survey," *First Monday*, vol. 5, no. 7, 2000. [Online]. Available: http://firstmonday.org/ojs/index.php/fm/article/view/769

[GPCFL04] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez, *Ontological Engineering – With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer, Jul. 2004.

[GR07] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[Gro14] "Grouplens," http://grouplens.org, 2014, accessed: 2014-04-21.

[Gru93] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[Hac14] "Hacker news," https://news.ycombinator.com, 2014, accessed: 2014-04-21.

[Har67] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i-iii part i. the basic model," *Management science*, vol. 14, no. 3, pp. 159–182, 1967.

[Hay09] C. Haythornthwaite, "Crowds and communities: Light and heavyweight models of peer production," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on.* IEEE, 2009, pp. 1–10.

[Hep07] M. Hepp, "Possible ontologies: How reality constrains the development of relevant ontologies," *Internet Computing, IEEE*, vol. 11, no. 1, pp. 90–96, 2007.

[HGM06] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Stanford InfoLab, Technical Report 2006-10, Apr. 2006. [Online]. Available: http://ilpubs.stanford.edu:8090/775/

[HGMR13] A. Halfaker, R. S. Geiger, J. T. Morgan, and J. Riedl, "The rise and decline of an open collaboration system: How Wikipedia's reaction to popularity is causing its decline," *American Behavioral Scientist*, vol. 57, no. 5, pp. 664–688, 2013.

[HJ02] C. W. Holsapple and K. D. Joshi, "A collaborative approach to ontology design," *Commun. ACM*, vol. 45, no. 2, pp. 42–47, Feb. 2002.

[HKB08] C. Hütter, C. Kühne, and K. Böhm, "Peer production of structured knowledge—an empirical study of ratings and incentive mechanisms," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management.* New York, NY, USA: ACM, 2008, pp. 827–842.

[HPZ06] N. Hu, P. A. Pavlou, and J. Zhang, "Can online reviews reveal a product's true quality?: Empirical findings and analytical modeling of online word-of-mouth communication," in *Proceedings of the 7th ACM Conference on Electronic Commerce*, ser. EC '06. New York, NY, USA: ACM, 2006, pp. 324–330. [Online]. Available: http://doi.acm.org/10.1145/1134707.1134743

[HRRK08] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan, "Predictors of answer quality in online q&a sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 865–874. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357191

[hto] "Historical Thesaurus of the Oxford English Dictionary," http://public.oed.com/historical-thesaurus-of-the-oed, accessed: 2014-04-01.

[HWC13] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods*. John Wiley & Sons, 2013, vol. 751.

[IPW10] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, ser. HCOMP '10. New York, NY, USA: ACM, 2010, pp. 64–67. [Online]. Available: http://doi.acm.org/10.1145/1837885.1837906

[J+07] R. Jäschke *et al.*, "Organizing publications and bookmarks in BibSonomy," in *CKC*, 2007.

[JB90] J. M. James and R. Bolstein, "The effect of monetary incentives and follow-up mailings on the response rate and response quality in mail surveys," *Public Opinion Quarterly*, vol. 54, no. 3, pp. 346–361, 1990.

[JF06] R. Jurca and B. Faltings, "Minimum payments that reward honest reputation feedback," in *Proceedings of the 7th ACM conference on Electronic commerce - EC '06*, 2006.

[JF07] ——, "Collusion-resistant, incentive-compatible feedback payments," in *Proceedings of the 8th ACM conference on Electronic commerce*, ser. EC '07. New York, NY, USA: ACM, 2007, pp. 200–209.

[JM09] D. Jurafsky and J. H. Martin, *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.

[KB06] C. Kühne and K. Böhm, "Classification schemes for contributions in peer-rating online communities – a quantitative assessment," in *Group Decision and Negotiation (GDN) 2006*, International Conference, Karlsruhe, Germany, June 25-28, 2006, Jun. 2006.

[KB14] ——, "Assessing the suitability of an honest rating mechanism for the collaborative creation of structured knowledge," *World Wide Web*, vol. 17, no. 1, pp. 85–104, 2014.

[KB15] C. Kühne and K. Böhm, "Protecting the Dawid–Skene algorithm against low-competence raters and collusion attacks with gold-selection strategies," *Social Network Analysis and Mining*, vol. 5, no. 1, 2015.

[KBY10] C. Kühne, K. Böhm, and J. Z. Yue, "Reviewing the reviewers: A study of author perception on peer reviews in computer science," in *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. IEEE, 2010, pp. 1–8.

[KKMF13] G. Kazai, J. Kamps, and N. Milic-Frayling, "An analysis of human factors and label accuracy in crowdsourcing relevance judgments," *Information retrieval*, vol. 16, no. 2, pp. 138–178, 2013.

[KKRK12] S. Kiesler, R. Kraut, P. Resnick, and A. Kittur, "Regulating behavior in online communities," in *Building Successful Online Communities: Evidence-Based Social Design*. The MIT Press, 2012, pp. 125–178.

[KMP10] S. Kochhar, S. Mazzocchi, and P. Paritosh, "The anatomy of a large-scale human computation engine," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, ser. HCOMP '10. New York, NY, USA: ACM, 2010, pp. 10–17.

[KPK09] A. Kittur, B. Pendleton, and R. E. Kraut, "Herding the cats: The influence of groups in coordinating peer production," in *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, ser. WikiSym '09. New York, NY, USA: ACM, 2009. [Online]. Available: http://doi.acm.org/10.1145/1641309.1641321

[KR12] R. E. Kraut and P. Resnick, "Encouraging contribution to online communities," in *Building Successful Online Communities: Evidence-Based Social Design*. The MIT Press, 2012, pp. 21–76.

[Kro90] D. A. Kronick, "Peer review in 18th-century scientific journalism," *Jama*, vol. 263, no. 10, pp. 1321–1322, 1990.

[Kun04] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[KV06] K. Kotis and A. Vouros, "Human-centered ontology engineering: The HCOME methodology," *Knowl. Inf. Syst.*, vol. 10, no. 1, pp. 109–131, Jul. 2006.

[KW93] S. J. Karau and K. D. Williams, "Social loafing: A meta-analytic review and theoretical integration." *Journal of personality and social psychology*, vol. 65, no. 4, p. 681, 1993.

[KWS03] L. I. Kuncheva, C. J. Whitaker, and C. A. Shipp, "Limits on the majority vote accuracy in classifier fusion," *Pattern Anal. Appl.*, vol. 6, no. 1, pp. 22–31, 2003.

[LGN73] M. R. Lepper, D. Greene, and R. E. Nisbett, "Undermining children's intrinsic interest with extrinsic reward: A test of the overjustification

hypothesis," *Journal of Personality and Social Psychology*, vol. 28, no. 1, p. 129, 1973.

[LIJ+14] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web Journal*, 2014.

[LK+77] J. R. Landis, G. G. Koch *et al.*, "The measurement of observer agreement for categorical data." *biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[LL02] E. A. Locke and G. P. Latham, "Building a practically useful theory of goal setting and task motivation: A 35-year odyssey." *American psychologist*, vol. 57, no. 9, p. 705, 2002.

[LM01] O. Lassila and D. McGuinness, "The role of frame-based representation on the Semantic Web," *Linköping Electronic Articles in Computer and Information Science*, vol. 6, no. 5, p. 2001, 2001.

[LR04] C. Lampe and P. Resnick, "Slash(dot) and burn: Distributed moderation in a large online conversation space," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '04. New York, NY, USA: ACM, 2004, pp. 543–550. [Online]. Available: http://doi.acm.org/10.1145/985692.985761

[LS97] L. Lam and C. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 27, no. 5, pp. 553–568, 1997.

[LT75] H. A. Linstone and M. Turoff, *The Delphi method: Techniques and Applications*. Addison-Wesley Publishing Company, Advanced Book Program Boston, MA, 1975.

[LW05] K. R. Lakhani and R. G. Wolf, "Why hackers do what they do: Understanding motivation and effort in free/open source software projects," in *Perspectives on Free and Open Source Software*. MIT press Cambridge, MA, 2005, pp. 3–22.

[LWVO10] C. Lampe, R. Wash, A. Velasquez, and E. Ozkaya, "Motivations to participate in online communities," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2010, pp. 1927–1936.

[LYZ13] H. Li, B. Yu, and D. Zhou, "Error rate analysis of labeling by crowdsourcing," in *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*, 2013.

[McG05] D. L. McGuinness, "Ontologies come of age," in *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2005, p. 171.

[Mik05] P. Mika, "Ontologies are us: A unified model of social networks and semantics," in *The Semantic Web–ISWC 2005*. Springer, 2005, pp. 522–536.

[MJD09] R. Meka, P. Jain, and I. S. Dhillon, "Matrix completion from power-law distributed samples," in *Advances in Neural Information Processing Systems*, 2009, pp. 1258–1266.

[MLMS06] A. D. Moor, P. D. Leenheer, R. Meersman, and V. Starlab, "Dogmamess: A meaning evolution support system for interorganizational ontology engineering," in *Proc. of the 14th International Conference on Conceptual Structures, (ICCS 2006*. Springer-Verlag, 2006, pp. 189–203.

[MM87] G. Marks and N. Miller, "Ten years of research on the false-consensus effect: An empirical and theoretical review," *Psychological Bulletin*, no. 102, 1987.

[MMM+11] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2857–2866.

[MP69] M. L. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.

[MPZJ07] N. H. Miller, J. W. Pratt, R. J. Zeckhauser, and S. Johnson, "Mechanism design with multidimensional, continuous types and interdependent valuations," *Journal of Economic Theory*, vol. 136, no. 1, pp. 476–496, 2007.

[MRS08] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[MRZ05] N. Miller, P. Resnick, and R. Zeckhauser, "Eliciting informative feedback: The peer-prediction method," *Management Science*, vol. 51, no. 9, pp. 1359–1373, Sep. 2005.

[MW70] A. H. Murphy and R. L. Winkler, "Scoring rules in probability assessment and evaluation," *Acta Psychologica*, vol. 34, pp. 273–286, 1970.

[Nat] "Can 'open peer review' work for biologist? Biology Direct is hopeful." [Online]. Available: http://www.nature.com/nature/peerreview/debate/op1.html

[Nat06a] "Nature's peer review debate," 2006. [Online]. Available: http://www.nature.com/nature/peerreview/debate

[Nat06b] "How can we get the best out of peer review? A recipe for good peer review," 2006. [Online]. Available: http://www.nature.com/nature/peerreview/debate/nature04995.html

[NCA08] N. F. Noy, A. Chugh, and H. Alani, "The CKC challenge: Exploring tools for collaborative knowledge construction," in *IEEE Intelligent Systems*, vol. 23, no. 1, 2008, pp. 64–68. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4444183

[Nov07] O. Nov, "What motivates Wikipedians?" *Commun. ACM*, vol. 50, no. 11, pp. 60–64, Nov. 2007. [Online]. Available: http://doi.acm.org/10.1145/1297797.1297798

[NP82] S. Nitzan and J. Paroush, "Optimal decision rules in uncertain dichotomous choice situations," *International Economic Review*, vol. 23, pp. 289–297, 1982. [Online]. Available: http://www.jstor.org/stable/2526438

[Obr03] L. Obrst, "Ontologies for semantically interoperable systems," in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 366–369. [Online]. Available: http://doi.acm.org/10.1145/956863.956932

[Ols65] M. Olson, *The logic of collective action public goods and the theory of groups.* Cambridge, Mass: Harvard University Press, 1965.

[PCL+07] R. Priedhorsky, J. Chen, S. T. K. Lam, K. Panciera, L. Terveen, and J. Riedl, "Creating, destroying, and restoring value in Wikipedia," in *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, ser. GROUP '07. New York, NY, USA: ACM, 2007, pp. 259–268. [Online]. Available: http://doi.acm.org/10.1145/1316624.1316663

[Pep00] S. Pepper, "The TAO of Topic Maps: finding the way in the age of infoglut," in *Proceedings of XML Europe*, 2000.

[PHT09] K. Panciera, A. Halfaker, and L. Terveen, "Wikipedians are born, not made: a study of power editors on Wikipedia," in *Proceedings of the ACM 2009 international conference on Supporting group work*. ACM, 2009, pp. 51–60.

[PL68] L. Porter and E. Lawler, *Managerial Attitudes and Performance*, ser. The Irwin-Dorsey series in behavioral science. Richard D. Irwin, 1968.

[PM04] H. S. Pinto and J. a. P. Martins, "Ontologies: How can they be built?" *Knowledge and Information Systems*, vol. 6, no. 4, Mar. 2004.

[PM13] M. Promberger and T. M. Marteau, "When do financial incentives reduce intrinsic motivation? comparing behaviors studied in psychological and economic literatures." *Health Psychology*, vol. 32, no. 9, p. 950, 2013.

[Pre04] D. Prelec, "A Bayesian truth serum for subjective data," *Science (New York, N.Y.)*, vol. 306, pp. 462–466, Oct. 2004.

[PRH+06] J. Pulido, M. Ruiz, R. Herrera, E. Cabello, S. Legrand, and D. Elliman, "Ontology languages for the Semantic Web: A never completely updated review," *Knowledge-Based Systems*, vol. 19, no. 7, pp. 489 – 497, 2006, creative Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705106000736

[R C13] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org

[Rei04] S. Reiss, "Multifaceted nature of intrinsic motivation: The theory of 16 basic desires," *Review of General Psychology*, vol. 8, no. 3, pp. 179–193, 2004.

[RF13] G. Radanovic and B. Faltings, "A robust bayesian truth serum for non-binary signals," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*, 2013.

[RGH77] L. Ross, D. Greene, and P. House, "The "false consensus effect": An egocentric bias in social perception and attribution processes," *Journal of Experimental Social Psychology*, vol. 13, no. 3, pp. 279–301, 1977.

[RHS06] J. A. Roberts, I.-H. Hann, and S. A. Slaughter, "Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects," *Management science*, vol. 52, no. 7, pp. 984–999, 2006.

[RJB04] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education, 2004.

[RLT+06] A. M. Rashid, K. Ling, R. D. Tassone, P. Resnick, R. Kraut, and J. Riedl, "Motivating participation by displaying the value of contribution," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 955–958.

[RMJ10] D. R. Raban, M. Moldovan, and Q. Jones, "An empirical study of critical mass and online community survival," in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '10. New York, NY, USA: ACM, 2010, pp. 71–80.

[RN13] S. Russell and P. Norvig, *Artificial Intelligence: Pearson New International Edition: A Modern Approach*, ser. Always learning. Pearson Education, Limited, 2013. [Online]. Available: http://books.google.de/books?id=DFJtngEACAAJ

[RY12] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 13, no. 2, 2012.

[Sav71] L. J. Savage, "Elicitation of personal probabilities and expectations," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 783–801, 1971.

[SB94] C. Science and T. Board, *Academic Careers for Experimental Computer Scientists and Engineers*. Washington, D.C.: National Academy Press, 1994.

[SBF98] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods," *Data & knowledge engineering*, vol. 25, no. 1, pp. 161–197, 1998.

[Sch06] S. Schaffert, "IkeWiki: A Semantic Wiki for collaborative knowledge management," in *Proceedings of the First International Workshop on Semantic Technologies in Collaborative Applications STICA 06*, R. Tolksdorf, E. P. B. Simperl, and K. Schild, Eds., 2006, pp. 388–396.

[SH08] K. Siorpaes and M. Hepp, "Games with a purpose for the Semantic Web," *IEEE Intelligent Systems*, vol. 23, no. 3, pp. 50–60, May 2008.

[SHC11] A. D. Shaw, J. J. Horton, and D. L. Chen, "Designing incentives for inexpert human raters," in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, ser. CSCW '11. New York, NY, USA: ACM, 2011, pp. 275–284.

[SKKM03] E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi, "An environment for distributed ontology development based on dependency management," in *International Semantic Web Conference*, 2003, pp. 453–468.

[SKW07] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.

[SLB09] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2009.

[SLR14] E. Simperl and M. Luczak-Rösch, "Collaborative ontology engineering: a survey," *The Knowledge Engineering Review*, vol. 29, pp. 101–131, 1 2014. [Online]. Available: http://journals.cambridge.org/article_S0269888913000192

[SOJN08] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08. Stroudsburg, PA, USA: Association

for Computational Linguistics, 2008, pp. 254–263. [Online]. Available: http://dl.acm.org/citation.cfm?id=1613715.1613751

[Sou05] A. Souzis, "Building a Semantic Wiki," *IEEE Intelligent Systems*, vol. 20, pp. 87–91, 2005.

[Spi02] R. Spier, "The history of the peer-review process." *Trends in biotechnology*, vol. 20, no. 8, pp. 357–8, Aug. 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12127284

[SS09] S. Staab and R. Studer, *Handbook on Ontologies*, 2nd ed. Springer Publishing Company, Incorporated, 2009.

[SS10] K. Siorpaes and E. Simperl, "Human intelligence in the process of semantic content creation," *World Wide Web*, vol. 13, no. 1-2, pp. 33–59, Mar. 2010.

[SSW09] F. M. Suchanek, M. Sozio, and G. Weikum, "SOFIE: a self-organizing framework for information extraction," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 631–640.

[ST06] E. P. B. Simperl and C. Tempich, "Ontology engineering: A reality check," in *OTM Conferences (1)*, 2006, pp. 836–854.

[Sta14] "Stack overflow help badges," http://stackoverflow.com/help/badges, 2014, accessed: 2014-04-21.

[SW01] B. Smith and C. A. Welty, "Fois introduction: Ontology - towards a new synthesis," in *Proceedings of the International Conference on Formal Ontology in Information Systems – FOIS*, 2001, pp. iii–ix.

[SW14] J. Solomon and R. Wash, "Critical mass of what? exploring community growth in WikiProjects," in *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICSWM)*, 2014.

[TEL+05] Y. a. Tijerino, D. W. Embley, D. W. Lonsdale, Y. Ding, and G. Nagy, "Towards ontology generation from tables," *World Wide Web*, vol. 8, no. 3, pp. 261–285, Aug. 2005.

[TN07] T. Tudorache and N. F. Noy, "Collaborative protege," in *CKC*, 2007.

[UG04] M. Uschold and M. Gruninger, "Ontologies and semantics for seamless connectivity," *ACM SIGMod Record*, vol. 33, no. 4, pp. 58–64, 2004.

[vA06] L. von Ahn, "Games with a purpose," *IEEE Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[VKV+06] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer, "Semantic Wikipedia," in *WWW*, 2006, pp. 585–594.

[VPTS05] D. Vrandecic, H. S. Pinto, C. Tempich, and Y. Sure, "The diligent knowledge processes," *J. Knowledge Management*, vol. 9, no. 5, pp. 85–96, 2005.

[VRGE+99] S. Van Rooyen, F. Godlee, S. Evans, N. Black, and R. Smith, "Effect of open peer review on quality of reviews and on reviewers' recommendations: a randomised trial," *Bmj*, vol. 318, no. 7175, pp. 23–27, 1999.

[Wel07] K. Weller, "Folksonomies and ontologies: Two new players in indexing and knowledge representation," in *Applying Web 2.0. Innovation, Impact and Implementation: Online Information 2007 Conference Proceedings*, London, 2007, pp. 108–115. [Online]. Available: http://wwwalt.phil-fak.uni-duesseldorf.de/infowiss/admin/public_dateien/files/35/1197280560weller009p.pdf

[WH07] D. M. Wilkinson and B. A. Huberman, "Assessing the value of cooperation in Wikipedia," *First Monday*, vol. 12, no. 4, 2007.

[Wie02] K. E. Wiegers, *Peer Reviews in Software: A Practical Guide*. Boston, MA.: Addison-Wesley, 2002.

[Wil08] D. M. Wilkinson, "Strong regularities in online peer production," in *Proceedings of the 9th ACM conference on Electronic commerce*. ACM, 2008, pp. 302–309.

[Win69] R. L. Winkler, "Scoring rules and the evaluation of probability assessors," *Journal of the American Statistical Association*, vol. 64, no. 327, pp. 1073–1078, 1969.

[WIP] J. Wang, P. G. Ipeirotis, and F. Provost, "Quality-based pricing for crowdsourced workers," NYU-CBA Working Paper CBA-13-06.

[WIP11] ——, "Managing crowdsourcing workers," in *The 2011 Winter Conference on Business Intelligence*, 2011, pp. 10–12.

[WKWea02] E. J. Weber, P. P. Katz, J. F. Waeckerle, and et al., "Author perception of peer review: Impact of review quality and acceptance on satisfaction," *JAMA*, vol. 287, no. 21, 2002.

[WLB12] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 20:1–20:36, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2333112.2333115

[WP12a] J. Witkowski and D. C. Parkes, "Peer prediction without a common prior," in *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM, 2012, pp. 964–981.

[WP12b] ——, "A robust bayesian truth serum for small populations." in *AAAI*, 2012.

[WP13] R. Weaver and D. Prelec, "Creating truth-telling incentives with the bayesian truth serum," *Journal of Marketing Research*, vol. 50, no. 3, pp. 289–302, 2013.

[WRW+09] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, 2009, pp. 2035–2043.

[XNH+11] Z. Xu, Y. Ni, W. He, L. Lin, and Q. Yan, "Automatic extraction of OWL ontologies from UML class diagrams: a semantics-preserving approach," *World Wide Web*, vol. 15, no. 5-6, pp. 517–545, Nov. 2011.

[YL10] H.-L. Yang and C.-Y. Lai, "Motivations of Wikipedia content contributors," *Computers in Human Behavior*, vol. 26, no. 6, pp. 1377–1383, 2010.

[Zac12] E. Zachte, "Wikimedia report card february 2012," 2012, accessed: 2014-04-01. [Online]. Available: http://stats.wikimedia.org/reportcard

[ZB07] V. Zacharias and S. Braun, "SOBOLEO – social bookmarking and lighweight engineering of ontologies," in *CKC*, 2007.