ABSTRACT

Title of dissertation: Diversity and Novelty: Measurement, Learning and Optimization

Faez Ahmed
Doctor of Philosophy, 2019

Dissertation directed by: Dr. Mark Fuge
Department of Mechanical Engineering

The primary objective of this dissertation is to investigate research methods to answer the question: "How (and why) does one measure, learn and optimize novelty and diversity of a set of items?" The computational models we develop to answer this question also provide foundational mathematical techniques to throw light on the following three questions: 1. How does one reliably measure the creativity of ideas? 2. How does one form teams to evaluate design ideas? 3. How does one filter good ideas out of hundreds of submissions? Solutions to these questions are key to enable the effective processing of a large collection of design ideas generated in a design contest. In the first part of the dissertation, we discuss key qualities needed in design metrics and propose new diversity and novelty metrics for judging design products. We show that the proposed metrics have higher accuracy and sensitivity compared to existing alternatives in literature. To measure the novelty of a design item, we propose learning from human subjective responses to derive low dimensional triplet embeddings. To measure diversity, we propose an entropy-based diversity metric, which is more accurate and sensitive than benchmarks.

In the second part of the dissertation, we introduce the bipartite b-matching problem and argue the need for incorporating diversity in the objective function for matching problems. We propose new submodular and supermodular objective functions to measure diversity and develop multiple matching algorithms for diverse team formation in offline and online cases. Finally, in the third part, we demonstrate filtering and ranking of ideas using diversity metrics based on Determinantal Point Processes as well as submodular functions. In real-world crowd experiments, we demonstrate that such ranking enables increased efficiency in filtering high-quality ideas compared to traditionally used majority voting.

Diversity and Novelty: Measurement, Learning and Optimization

by

Faez Ahmed

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Dr. Mark Fuge, Chair/Advisor
Dr. Shapour Azarm
Dr. Jennifer Golbeck
Dr. Linda Schmidt
Dr. John Dickerson

“If you can’t measure it,

you can’t improve it.”

Acknowledgments

First and foremost, I thank God for bestowing upon me innumerable gifts, many more than what I desired or deserved. I am privileged to have a loving family, good friends, and amazing mentors, who have helped me at different steps. I thank all the people who have been instrumental in the completion of this dissertation and helped me prepare for the next steps in life.

I would like to thank my advisor, Dr. Mark Fuge for the countless hours he spent discussing new ideas, exploring the right approach to problems, and patiently giving me advice. In the past four years, I have appreciated how much he cares about his students. It is a pleasure to work with such an extraordinary individual, who is a rare mix of brilliance, enthusiasm, and magnanimity. I would also like to thank my collaborator, Dr. John Dickerson. Without his extraordinary ideas and theoretical expertise, this thesis would have been a distant dream. Thanks are due to Dr. Shapour Azarm, Dr. Linda Schmidt, and Dr. Jennifer Golbeck for agreeing to serve on my thesis committee and giving me insightful feedback. I am reminded that many (my mentors, collaborators, and reviewers) have quietly enabled me without asking for acknowledgment. I am grateful and will work to pay it forward.

My friends and colleagues at UMD have enriched my graduate life in many ways and deserve a special mention. I would like to express my gratitude to Saurabh, Shashank, Lovlesh, Kiranraj, Abhinav, Sarthak and Wei for their friendship and support. It is impossible to remember all, and I apologize to those I have inadvertently left out.

I owe my deepest thanks to my parents and siblings, who have always supported me

and guided me on multiple occasions. Words cannot express my gratitude. Lastly, I am forever in debt to my wife for her understanding, endless patience, and encouragement when it was most needed. She inspires me to become a better person.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BoL	Bag-of-Lemons
CAT	Consensual Assessment Technique [122]
D-WBM	Diverse Weighted Bipartite Matching
DBLemons	Diverse Bag-of-Lemons
DPP	Determinantal Point Processes
EG	Entropy Gain
GD-WBM	Greedy Diverse Weighted Bipartite Matching
GNMDS	Generalized Non-metric Multidimensional Scaling [5]
HHI	Herfindahl–Hirschman–Hirschman Index [125]
HHID	Herfindahl–Hirschman–Index for Design
MIQP	Mixed Integer Quadratic Program
MSE	Mean Squared Error
NDCG	Normalized Discounted Cumulative Gain
NM	Design variety metric proposed by Nelson *et al.* [193]
PoD	Price of Diversity
RKHS	Reproducing Kernel Hilbert Space
SVS	Design variety metric proposed in Shah *et al.* [235]
SVS_n	Design novelty metric proposed in Shah *et al.* [235]
TF-IDF	term frequency-inverse document frequency
WBM	Weighted Bipartite Matching

Chapter 1: Introduction

Merriam-Webster [1] defines "novelty" to mean " something new or unusual". It defines "diversity" as "the condition of having or being composed of differing elements". The objective of this dissertation is to investigate research methods to answer the question: *How (and why) does one measure, learn and optimize novelty and diversity of a set of items?*

Measuring novelty and diversity is needed in many domains like Engineering Design, Signal Processing and Machine Learning. In Engineering Design, one is often tasked to select the most creative idea from a large collection of submissions. An integral part of measuring the creativity of ideas is understanding how unique or novel each idea is. Novelty measurement is also important in Signal Processing and Machine Learning domains, for tasks like outlier detection or recommending unique items to people. Diversity measures how well a design space is explored, which often correlates with the success of the final product in the field of Product Design. Diversity optimization is also employed in finding a subset of any dataset, which can lead to faster training of machine learning models. In organizations, teams with diverse skill sets are often sought after. For all these applications, developing ways to measure and optimize diversity is required.

Despite their ubiquity, measuring, learning and optimizing both these metrics is non-trivial and present complex mathematical and practical challenges. Diversity is of-

ten measured using submodular coverage functions [171] (which are NP hard to maximize [47]), while novelty is either measured subjectively or often interpreted as a measure of outlier detection [211] (which requires advanced statistical tools). The key difficulties in answering our primary question in any generalizable, predictable way are:

- There is no consensus on what method to use for measuring novelty of design items (*e.g.* subjective vs quantitative).
- There is a lack of standard benchmark datasets, which makes learning novelty from past data difficult.
- Information about attributes are often not available for items like design sketches and it is non-trivial to pin-point what attributes should be used in computational methods.
- Multiple methods (like Shannon entropy [236], Determinantal Point Processes [162]) exist to measure diversity but it is not well understood which method is more suitable and under what conditions.
- Finding sets of items with maximum diversity is an NP-hard problem [1] which means they cannot be solved in polynomial time.

Before going further into what we did to address these gaps, we would like to first discuss the broader application which this dissertation targets. While the methods developed in this dissertation apply widely to many domains, we select online design contests as our primary application. By design, we mean a plan or specification in the

[1] It is a specific case of submodular maximization with matroid constraints [48].

form of a prototype, product or process. It can refer to sketches of physical products, CAD models of a machinery, a text document describing a process or an idea. Online design contests are often conducted by organizations (*e.g.* OpenIDEO [97], GrabCAD [225]) to gather ideas from many geographically distributed people. These contests have been widely adopted by industry (Google, IBM, Local Motors, GE, BMW *etc.*) as well as government agencies (*e.g.* DARPA). To help improve these contests, these organizations need methods to assess creativity of a large collection of ideas, form teams of reviewers to evaluate these ideas and finally find good ideas which should be funded or implemented.

In this dissertation, we develop ranking, matching, and creativity estimation models to address parts of these problems. Although, the primary objective of this dissertation is to investigate research methods to answer the question: "How (and why) does one measure, learn and optimize novelty and diversity of a set of items?", as we show later that answering this primary question allows us to to address some of the practical gaps which occur in conducting online design contests. By developing ways to optimize diversity in discrete space, we show that we can form diverse teams, which can help design contest organizers in allocating judges to ideas. By developing diversity metrics in continuous space, we show ways to filter a set of diverse ideas. This enables design contest organizers to find a diverse set of high quality ideas from a large collection. Finally, by finding ways to measure and learn novelty, we develop explainable metrics, which can be used by judges to evaluate ideas submitted for a design contest. Hence, by answering our primary question about novelty and diversity metrics, each of our chapters throws light on the following three secondary questions:

1. Chapter 2: How does one reliably measure the creativity of ideas?

2. Chapter 3: How does one form teams to evaluate design ideas?

3. Chapter 4: How does one filter good ideas out of hundreds of submissions?

In addressing these secondary questions in each chapter, we address the central theses. We develop principled methods for measurement, learning, and optimization of novelty and diversity metrics and address both theoretical and practical gaps in existing literature. The mathematical models we develop also provide foundational techniques for other fields where similar questions arise, such as Machine Learning (*e.g.* in submodular optimization), information retrieval (*e.g.* to filter content), and CAD (e.g., 3D geometric search/creation). Below, we provide a brief summary of each secondary question and how they tie to our primary question.

Chapter 2 Summary: How does one reliably measure the creativity of ideas?

After successful completion of a design contest, many possible ideas are collected and one needs to search for creative solutions among many possible options. However, giving a numerical score to measure creativity is difficult due to two main reasons: 1. creativity measurement requires human insight, which is hard to quantify; and 2. mathematical functions to compute scores cannot be applied to designs, which are not in a vector space a priori. As a result, most existing creativity measures are criticized for being subjective and not generalizable to new domains. To overcome these problems, we proposed a new framework to study novelty and diversity (metrics related to creativity) using modern machine learning methods [16, 17]. In the first part of this chapter, we develop a way

to measure novelty of ideas which are not easily represented in vector space like design sketches. We first find design embedding in 2-D Euclidean space, such that the embedding captures human intuition of relationships between items. We find these relationships by asking subjective pairwise comparisons from people. Next, we calculate an embedding using kernel learning methods and then use a family of mathematical functions to measure novelty. In the second part of this chapter, we show that a new metric to measure diversity, based on Herfindahl–Hirschman Index (HHI) [217, 125] is more accurate and sensitive than the alternatives. Overall, this chapter provides data-driven methods to measure and optimize design metrics.

Chapter 3 Summary: How does one form teams to evaluate design ideas?

Assuming an organization has conducted a design contest, to process idea submissions meaningfully, it needs to find teams of reviewers who can evaluate these ideas. This team formation problem is often referred as the bipartite matching problem, which is a a classical and long-standing problem in computer science and economics, with widespread application in health-care, education, advertising, and general resource allocation. In bipartite matching, items on one side of a market are matched to items on the other. For example, allocating conference papers to reviewers requires allocating multiple reviewers to each paper and each paper should be matched to three or more reviewers

Bipartite matching can also be used to form teams to review ideas, by allocating reviewers to ideas based on their expertise. However, this traditional model assumes that each person is independent of their teammates, which is often not the case. Having multiple reviewers who all have the same skillset or are similar to each other may not

be desirable, as they may provide similar feedback. To address this, we proposed new algorithms for diverse matching [9] problems that balance diversity and quality of the solution. We demonstrated the efficacy of our methods on three real-world datasets and showed that, in practice, encouraging diverse teams does not sacrifice performance. We also proposed a method for the online version of this problem [12], where we do not know the availability of people beforehand. In such cases, people arrive one at a time and we need to decide at the moment whether to assign a newly arrived person to a team or not. The online matching algorithms are relevant to crowd markets, reviewer assignment for journal papers as well as domains where discrete items arrive sequentially. Overall, this chapter contributes new algorithms for forming diverse teams and provides performance guarantees of these algorithms.

Chapter 4 Summary: How does one filter good ideas out of hundreds of submissions?

In many online contests, the end goal is to fund the implementation of a small subset of novel and useful ideas. Assuming we know the quality of all ideas and their representation in vector space, in this chapter, we propose algorithms to filter few good ideas from a large collection. This problem can be formulated as constrained submodular optimization, which is NP-Hard [48].

In the first part, we proposed and successfully implemented an optimization method [15] to filter out a small subset of ideas which has both high quality and coverage (ideas are different from each other). We also proposed a diverse ranking [14] method to rank

order all ideas by maximizing quality and coverage metrics. We showed the applicability of our algorithms to information retrieval and finding design analogies. To rank ideas, we showed how to capture the similarity between items, modeled coverage with Determinantal Point Processes and proposed an efficient combinatorial optimization method to find the solution. To test diverse ranking algorithms in the real world, we deployed it on a web-platform [181] with the aim of searching for best ideas using crowd voting. Our results on two real-world problems (OpenIDEO and UNICEF) showed how diverse ranking leads to a large increase in filtering efficiency.

1.1 Impact of This Dissertation

This dissertation tackles problems central to engineering challenges noted by the NSF. Three of the NSF's 10 big ideas pertinent to this work are: 'Future of Work at the Human-Technology Frontier,' 'Growing Convergence Research,' and 'Harnessing the Data Revolution.' The NSF planned to spend $\$30$ million in 2019 on these ideas. This work impacts other domains too. First, our methods help improve 'Fairness' and 'Innovation capability' in idea evaluation by proposing accurate metrics, which ensure that every idea is evaluated and no idea falls through the cracks. Companies like OpenIDEO and GrabCAD, who regularly conduct design contests can adopt metrics proposed in our work to evaluate ideas and teams. Second, typical matching algorithms find teams such that the total utility of the matching is maximized. This utility is often learned from past data, which may have some biases. Our diverse matching algorithms encourage all classes to be represented in teams by facilitating diverse group formation. Using these algorithms,

one can form teams balanced for any quality like gender, race or skillsets. Our algorithms can be applied for admissions in different universities, to hire a diverse cohort and match students to advisors. Our results on Price of Diversity, showing how little utility is lost for large gains in diversity provide a quantitative way to have discussions around benefits of diversity and directing policies accordingly. Finally, these methods have a significant economic impact too. Millions of dollars and thousands of person-hours are spent each year in trying to improve economic competitiveness via better innovation and design practices. Data from four large firm idea contests [154] indicates that a firm appealing to the crowd will receive between 46K - 460K ideas (IBM's Innovation Jam received 46K ideas, Dell's IdeaStorm 20K ideas, Google's 10^{100} project 150K ideas and Singapore Thinathon's contest 454K ideas). Past research [220] also indicates that the cost for an expert to evaluate a single idea in a Fortune 100 company is $500, with an average effort of 4 hours. We demonstrate from our results how we can help reduce this high cost of evaluating a large collection of ideas by proposing metrics which are an accurate, more reliable, faster and better reflection of human insight.

1.2 How to Use This Dissertation?

This dissertation is structured in three chapters, each addressing one question. We believe that different parts of this dissertation may interest difference audiences. Our suggestions regarding the same are provided below:

Designers looking to understand the creativity process. Chapter 2 provides us insights into how people reason about creativity. Designers who are actively involved in creative

tasks and are interested in decoding how they think about creativity can read our findings on idea maps.

Online community managers interested in filtering ideas. Chapter 4 provides both theory and application to filter ideas. Community managers who have just conducted a design contest and want to find a high quality diverse ranking of ideas can download our code and use it on their dataset. On the other hand, online community managers who are about to conduct a new design contest can take cues from our ranking methods, which is shown to enable more efficient crowd filtering.

Design Methodology researchers looking to use better metrics. In chapter 2, we outlined desirable properties for design metrics and suggest new alternative metrics (like HHI and embedding based metrics) which are shown to be more accurate than the metrics commonly used in literature. Design Methodology researchers who study creativity metrics may find our findings useful. Researchers would also find the methods outlined in chapter 4 useful to evaluate a large collection of ideas collected in a classroom setting.

Computer Scientists looking to apply their algorithms to new domains. Chapter 2 discusses the complexities of problems in Engineering Design domain (like hierarchical dependencies) and applies existing triplet embedding algorithms on human responses. We have made few annotated datasets public, which can help other Computer Scientists to address new problems in Engineering Design by applying their algorithms.

Computer Scientists looking to develop new theory. Chapter 3 introduces new algorithms to perform diverse matching for bipartite graphs. Computer scientists who are working in reducing the time or space complexity of diverse matching algorithms may read section 3.5.3 on our approach to the problem using auxiliary graphs or section 3.4.2 for understanding how we formulated diverse matching as a mixed integer quadratic problem.

1.3 Publications Related to This Dissertation

A subset of my work has been published in journals (JMD) and conferences (IJCAI, CSCW, IDETC). Below is a list of publications that support different chapters:

Chapter 2

- Faez Ahmed, Mark Fuge, Sam Hunter, and Scarlett Miller. Unpacking subjective creativity ratings: Using embeddings to explain and measure idea novelty. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T06A003–V007T06A003. American Society of Mechanical Engineers, 2018

- Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Interpreting idea maps: Pairwise comparisons reveal what makes ideas novel. volume 141, page 021102. American Society of Mechanical Engineers, 2019

- Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Measuring and optimizing design variety using herfindahl index. In

ASME International Design Engineering Technical Conferences, Anaheim, USA, August 2019. ASME

Chapter 3

- Faez Ahmed, John P Dickerson, and Mark Fuge. Diverse weighted bipartite b-matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 35–41. AAAI Press, 2017

- Faez Ahmed, John P Dickerson, and Mark Fuge. Online diverse team formation. In *Under submission*, 2019

- Saba Ahmadi, Faez Ahmed, John P Dickerson, Mark Fuge, and Khuller Samir. On diverse bipartite b-matchings. In *Under submission*, 2019

Chapter 4

- Faez Ahmed and Mark Fuge. Capturing winning ideas in online design communities. In *20th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, Portland, USA, February 2017. ACM

- Faez Ahmed, Mark Fuge, and Lev D. Gorbunov. Discovering diverse, high quality design ideas from a large corpus. In *ASME International Design Engineering Technical Conferences*, Charlotte, USA, August 2016. ASME

- Faez Ahmed and Mark Fuge. Ranking ideas for diversity and quality. *Journal of Mechanical Design*, 140(1):011101, 2018

- Ioanna Lykourentzou, Faez Ahmed, Costas Papastathis, Irwyn Sadien, and Konstantinos Papangelis. When crowds give you lemons: Filtering innovative ideas using a diverse-bag-of-lemons strategy. In *21st ACM Conference on Computer-Supported Cooperative Work & Social Computing*, Jersey City, USA, November 2018. ACM

1.4 Structure of This Dissertation

This document is divided into five chapters. Chapter 1 provides an executive summary overview of the document. Chapter 2 focuses on design metrics (novelty and diversity) in engineering design. There, we undertake two research tasks on measuring the novelty of design items using pairwise comparisons and proposing a new metric to measure the diversity of ideas. Chapter 3 introduces diversity in matching. There, we define the problem of diverse matching and propose algorithms for solving offline and online diverse matching problem. Chapter 4 proposes algorithms to filter good ideas using diverse idea selection and filtering. In the first part, we propose multi-objective diverse ranking methods, while the second part shows how a diverse ranking method can improve filtering efficiency for design contests. Finally, Chapter 5 summarizes the dissertation.

Chapter 2: Data-driven Design Metrics: How does one reliably measure the creativity of ideas?

Creativity is the driving force of innovation in the design industry. Despite many methods to help designers enhance the creativity of generated ideas, not much research has focused on what happens after this generation [242]. One of the main problems that design managers face after the completion of an ideation exercise is how to judge the submitted ideas. Contributors have just sent in a flood of design ideas of variable quality, and these ideas must now be reviewed, in order to select the most promising among them. Idea evaluation has been highlighted as a central stage in the innovation process in fields like design and management [115]. However, many of the existing methods of idea evaluation are inherently subjective. An emerging thread of research within idea evaluation is on attempts to quantitatively assess quality, novelty and variety (key components of creativity measurement) of ideas [179, 228, 139]. In this chapter, we investigate rigorous and verifiable computational methods for measuring the novelty and variety (which is often referred as diversity in other domains) of engineered systems using theories from engineering design, cognitive psychology, and applied mathematics. Specifically, we divide our work into two research tasks:

- Research Task 1: Data-driven Novelty Metrics: In the first research task, we inves-

tigate data-driven novelty metrics by learning similarity preferences from people. Assessing the similarity between design ideas is an inherent part of many design evaluations to measure novelty. In such evaluation tasks, humans excel at making mental connections among diverse knowledge sets to score ideas on their uniqueness. However, their decisions about novelty are often subjective and difficult to explain. In this chapter, we demonstrate a way to uncover human judgment of design idea similarity using two-dimensional idea maps. We derive these maps by asking participants for simple similarity comparisons of the form "Is idea A more similar to idea B or to idea C?" We show that these maps give insight into the relationships between ideas and help understand the design domain. We also propose that novel ideas can be identified by finding outliers on these idea maps. To demonstrate our method, we conduct experimental evaluations on two datasets — colored polygons (known answer) and milk-frother sketches (unknown answer). We show that idea maps shed light on factors considered by participants in judging idea similarity and the maps are robust to noisy ratings. We also compare physical maps made by participants on a white-board to their computationally generated idea maps to compare how people think about the spatial arrangement of design items. This method provides a new direction of research into deriving ground truth novelty metrics by combining human judgments and computational methods.

- Research Task 2: Data-driven Variety Metrics: In the second research task of this chapter, we propose a new design variety metric based on the Herfindahl index. We also propose a practical procedure for comparing variety metrics via construct-

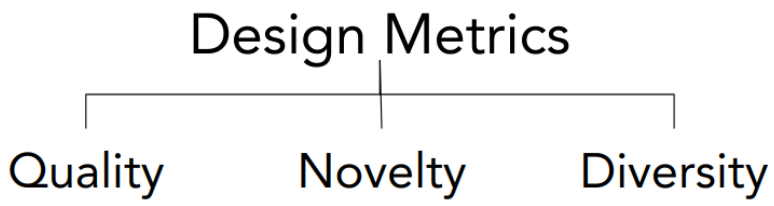

Figure 2.1: This dissertation mainly focuses on two main design metrics — Novelty and Diversity. Note that Novelty and Quality are often combined to measure Creativity. Diversity is also referred to as Variety. Other metrics like Usefulness, Surprise, Functionality, Feasibility, Impact, Investment potential, Scalability etc. *are not directly studied in this dissertation, although they often relate to creativity measurement.*

ing ground truth datasets from pairwise comparisons by experts. Using two new datasets, we show that this new variety measure aligns with human ratings more than some existing and commonly used tree-based metrics. This metric also has three main advantages over existing metrics: a) It is a super-modular function, which enables us to optimize design variety using a polynomial time greedy algorithm. b) The parametric nature of this metric allows us to fit the metric to better represent variety for new domains. c) It has higher sensitivity in distinguishing between the variety of sets of randomly selected designs than existing methods. Overall, our results shed light on some qualities that good design variety metrics should possess and the non-trivial challenges associated with collecting the data needed to measure those qualities.

2.1 Background and Motivation

In this section, we provide background about existing creativity and coverage metrics and highlight the key reasons which motivate our research.

2.1.1 Creativity Metrics: Quality and Novelty

Creativity is the capacity to generate unique and original work that is useful [23, 244, 192]. Creativity is useful at both individual and societal levels. At the individual level, creativity helps in effectively solving day-to-day tasks. At a societal level, it can yield meaningful scientific findings [244]. The creativity of ideas is often viewed as the comparison of design ideas for quality and novelty. Quality is a measure of the designs' performance [183] and can be defined using multiple domain dependent factors like functionality, feasibility, usefulness, impact, investment potential, scalability, *etc.* In contrast to quality, novelty represents the uniqueness of an idea or how different it is from other designs in its class[267]. Merriam-Webster [1] defines "novelty" to mean "new and not resembling something formerly known or used". Novelty measurement in design is the task of evaluating design items that differ in some respect from all other items. Both these metrics can be measured by human judges or using automated methods.

When ideas are being judged by humans, the judges may be categorized as experts or non-experts. Experts have substantial knowledge of the field and of the market, and can thus provide more informed and trustworthy evaluations [62]. Many crowdsourcing platforms such as Topcoder, Taskcn, and Wooshii use expert panels to select contest winners [61]. However, experts are also scarce and expensive, since gaining expertise on a

particular innovation subfield takes a substantial amount of training. As an alternative to expensive experts, crowds have been proposed [108] to evaluate ideas due to their large diversity of viewpoints, knowledge and skills [249]. However, there is no clear evidence demonstrating that crowds can be used as a proxy for experts' evaluations to assess a large number of ideas [106].

In contrast to subjective assessment by human judges, automated methods apply fixed rules to judge ideas represented by a set of design attributes [1] Objective methods of novelty measurement in Engineering design (like SVS [234]) are criticized for lack of generalizability. In the Research Task 1, we try to address some of these gaps found in existing novelty metrics by proposing a method combining the subjective and objective methods.

2.1.2 Coverage Metrics: Variety or Diversity

A well-known outlook relates creativity with divergent thinking — the capacity to produce a wider variety of ideas with higher fluency. Divergent thinking has been shown to correlate with the success of the final product [258, 3, 90, 34]. Prior work supports that chances of solving a problem increase when a more diverse set of ideas is produced in the initial stages of the design process [235, 201, 85]. These findings encourage the need to explore the design space in the early stages of design [121]. But how does one quantify

[1]Note that we use the words 'attribute' and 'feature' interchangeably throughout text as these words are commonly used in engineering design and computer science. They both refer to a vector quantifying characteristic traits of an item *e.g.*a rectangle with sides of unit length and two units breadth can be written as [1, 2] to characterize its length and width.

design space exploration?

Engineering researchers have sought to capture how "explored the solution space" is by measuring design variety (pg. 117, [235]). There are two approaches typically deployed in engineering literature to measure design variety: subjective ratings of variety and a genealogical tree approach. As one example of subjectively evaluating design variety, Linsey *et al.* [174] proposed taking a set of ideas and dividing them into pools based on intuitive categories created by the coder. The metric relies on a rater's mental model rather than a quantitative procedure [235]. While these subjective ratings provide a relatively efficient method for measuring design variety in terms of the amount of time and effort required to code design variety, this efficiency comes at the potential cost of the validity and reliability of the metric.

In contrast to subjective ratings, the other approach to measure design variety is using a genealogical tree approach. In these approaches, subjective human raters are replaced with a deterministic formula that depends on the attributes of a set of designs. One of the first metrics to use this approach was developed by Shah, Smith and, Vargas-Hernandez [235] (SVS metric) who broke each design into four hierarchical levels (physical principle, working principle, embodiment, and detail) to calculate design variety. The SVS metric is repeatable and attempts to reduce subjectivity by using predefined criteria for measuring variety. However, many researchers have criticized it due to its lack of sensitivity and accuracy. For example, the genealogical tree calculation method (like SVS) has been shown to be inconsistent with experts ratings of variety [175]. In addition, studies have shown that the sensitivity of the SVS metric diminishes when it is applied to large datasets [240] due to the exclusion of important abstract differences and generally

focuses on dissimilarity in the embodiment level [209].

In this chapter, we reexamine these hierarchical metrics and compare them to methods of calculating diversity from other (non-engineering) domains. Specifically, we compare the tree-based measures of SVS [235] and NM [193] with the Herfindahl–Hirschman Index (HHI), which is a statistical measure of concentration [217, 125]. The HHI accounts for the number of firms in a market, as well as their concentration, by incorporating the relative size (that is, market share) of all firms in a market. HHI is used by the Department of Justice and the Federal Reserve in the analysis of competitive effects of mergers. The key idea behind this metric is that market with more concentration will have a few large square terms. By comparing HHI to SVS [235] and NM [193], this work argues and empirically demonstrates that HHI is a more accurate measure for variety that has clear benefits for engineering and design measurement applications.

2.2 Literature Review

In this section, we review research related to novelty computation, which relates to our work. We first discuss qualities of a good metric followed by creativity and variety rating methods used in design.

2.2.1 Qualities of a Good Metric: Repeatability, Validity and Explainability

Quality control is essential when creating and evaluating metrics that map abstract concepts like creativity to quantitative measures. Particularly when metrics can be either

subjective and objective in scientific research, we need to demonstrate both the reliability and validity of such metrics without circularity [237], as well as reduce subjectivity in measurement techniques. For example, in the field of psychometrics, researchers try to craft sets of questions that produce internally consistent results — that is, if one asks the same questions one should get repeatable, similar answers even under minor changes to the test environment or experimental setup [156]. However, this only implies repeatability and not validity. Validity refers to the extent to which a measurement reflects the absolute state of an artifact under observation — the ground truth). The term "valid" implies an external frame of reference or a universally accepted standard against which a measurement is tested [260]. There is a wide range of creativity metrics that leverage a rater's expertise in a given domain to ensure metric validity. This is necessary to eliminate circularity or measuring unvalidated metrics against other unvalidated metrics [116].

The key assumption in the past research is that raters who have considerable experience in a given domain are best suited to provide a ground truth for tasks like evaluating creativity. We obtain this ground truth from real-world human evaluations, which can be used to measure the accuracy of any new metric. However, only using experts is no panacea. Expert time and effort is a scarce commodity, and this forces researchers to develop objective metrics that can aid quasi-experts or novice raters in accurately evaluating processes and ideas. The central hypothesis of that past work (which this work also shares) is that by validating objective metrics against expert raters, the joint predictive power of expertise and repeatable objective research methods will outperform either by themselves.

Metrics used to measure variety like SVS and NM, aim to reduce subjectivity on

the rater's part, to increase robustness in the processes used to analyze designs. When a metric is created, it is important to establish some desiderata (qualities we want) and acceptable qualities the metric must possess to ensure we obtain reliable results upon its execution. One example of establishing acceptable qualities of a metric was the work of Simonton and Amabile [24], who were key in standardizing the measurement of creativity in psychological research. Previously, most methods utilized pencil and paper tests, personality tests, biographical inventories (such as Schaefer and Anastasi's biographical inventory [230] and Taylor's Alpha Biographical Inventory [255]) and behavioral tests such as Torrance Tests of Creative Thinking. These tests were debatable in experiments that sought to reduce within-group variability and generally lacked a clear creativity definition and an effective strategy to avoid biases on behalf of the rater [24].

Good metrics are required to have the ability to establish ground truths using expert agreements and must be replicable by other raters who use the metric. For subjective metrics, high inter-rater reliability and internal consistency are some of the desired qualities of the metric [70]. We argue that for any new design metric, repeatability, validity, and explainability are also desirable qualities. If ground truth estimates of a quantity are available, then a new metric should align with this ground truth and the measurements should be repeatable. Metrics should also give explainable scores, that is, it should be possible to explain why one set of designs received a higher score than another set using a given metric. In judging existing metrics or proposing new design metrics, our goal is to test them for repeatability, validity and explainability.

2.2.2 Creativity Ratings

In the social sciences, creativity is often measured subjectively through the Consensual Assessment Technique (CAT) [122]. They define a creative idea as something that experts in the idea's or project's focus area independently agree is creative. CAT is considered one of the gold standards for creativity assessment as it can reliably assess creativity, through the consensual assessments of domain experts. However, it is difficult to explain what factors are used by experts to give a particular creativity or novelty scores to ideas. As humans have limited memory, it is also possible that while judging novelty of every design, experts may not remember all existing designs similar to it or they underestimate the originality of truly novel ideas [170]. By using different attributes or different criteria of evaluation within the same attribute, it is possible that experts will decide on completely different "novel" items. Novelty assessment is necessary in granting patents [232], as the patent legally restrains other people from exploiting the invention. For patent filing, novelty of an invention is assessed by a search through the prior art in the relevant technical field. A prior art search is generally performed with a view to proving that the invention is "not new" or old. While it is impossible to find all past sources of knowledge, a prior art search is often performed by using a keyword search of large patent databases, scientific papers and publications, and on any web search engine. In computational terms, the prior art search creates a set of related items and then novelty is judged based on dissimilarity of the idea requesting patent from all items in the set.

In contrast, engineering design creativity research focuses on the measurable aspects of an idea by breaking down the concept into its different components and measur-

ing their creativity in various ways [234, 268, 139]. For example, one of the commonly used tree-based metrics [234] breaks down creativity into quantity, quality, novelty, and variety. These methods are widely adopted in engineering due to limited rater bias [199]. The resultant novelty score of an idea depends on which attributes are considered in the tree and may vary between two different raters or trees[42]. Despite the existence of multiple metrics in engineering design for measuring design creativity, most methods have been heavily criticized for their lack of generalizability across domains, the subjectivity of the measurements and the timeliness of the method for evaluating numerous concepts [31, 51]. Computational methods also exist to measure novelty of items if their feature vectors [211] or attributes are available. However, when we do not know any attribute of the design in question (hence, cannot represent the design in vector space), computational methods cannot be applied. In this research task, we address this problem, where design attributes are not available or are partially available. We discuss how ordinal comparisons by human raters help us find attributes via design embeddings, which facilitate visualization of space as well as enable novelty computation.

2.2.3 Design Embeddings from Ordinal Comparisons

In a typical novelty detection method, one assumes to be given a set of items together with feature vectors (attributes) or a similarity function quantifying how "close" items are to each other [211]. However, a natural question that arises is, what happens when such attributes are not available or they are difficult to quantify. For example, it may be difficult to provide attributes for a collection of art images. Similar difficulty can be seen for design ideas which often lack compact vector representations or known similarity mea-

sures. One solution to this problem is to directly ask people about how similar ideas are and try to estimate the design embedding (attributes) which humans may have considered in deciding their subjective responses.

There are two common ways to collect similarity ratings from people. In the first way, one typically asks people to rate the perceived similarity between pairs of stimuli using numbers on a specified numerical scale (such as a Likert scale) [167]. Methods like classical multi-dimensional scaling [257] can be used with these ratings to find an embedding. However, these ratings are not considered suitable for human similarity judgments as different raters use different "internal scales" and raters may be inconsistent in their grading [263].

As humans are better at comparing items than giving absolute scores [245], the second way is to gather ordinal judgments. For instance, triplet ratings consist of asking subjects to choose which pair of stimuli out of three is the most similar in the form "Is A more similar to B or to C?". Once similarity judgments are captured, one can use a number of machine-learning techniques that try to find an embedding that maximally satisfies those triplets. Examples of such techniques include Generalized Non-metric Multidimensional Scaling (GNMDS) [5], Crowd Kernel Learning [251] and Stochastic Triplet Embedding [263]. Such methods take input of triplet ratings and output either an embedding or similarity kernel between items which best satisfy human triplet responses. The resultant embedding have been used for visual exploration of design space [273], and we propose that they can be used for novelty computation too.

Motivated by the fact that humans essentially think in two or three dimensions, many methods to visualize high dimensional data by mapping it to lower dimension mani-

folds have been studied extensively [253, 182]. Design space exploration techniques [218] have been developed to visualize a design space and generate feasible designs. Low dimensional embeddings obtained using ordinal comparisons can also help in better understanding the decision making of raters by visualizing the design space.

Techniques for capturing similarity among items using triplets have been applied in many areas like computer vision [226], sensor localization [197], nearest neighbor search [114] and density estimation [261]. In [78], authors learn perceptual kernels using different similarity methods. They find that triplet matching exhibits the lowest variance in estimates and is the most robust across the number of subjects compared to pairwise Likert rating and direct spatial arrangement methods. Siangliulue *et al.* [238] use triplet similarity comparisons by crowd workers to create spatial idea maps. They show that human raters agree with the estimates of dissimilarity derived from idea maps and use those maps to generate diverse idea sets. Our work in Research Task 1 differs from their work as our goal is to use idea maps to measure the novelty of design ideas and uncover what factors are used by raters in deciding similarity between items. Next, we dive into reviewing past work on design variety metrics.

2.2.4 Design Variety and its Importance

The measure of design variety in engineering was introduced as a means to measure how well someone explores the solution space during a design task [86]. The measure of design variety is important because research has shown that "there is no way to generate an optimum solution without exploring the solution space through early tentative ideas" (Pg.11 [71]). Generating a large number of ideas with iterative or small changes does not

result in effective concept generation or innovative products. Hence, the potential to develop ideas of broad variety is correlated with the ability to successfully reconstruct and solve problems. This ability is referred to as cognitive restructuring in psychology [235] which has been used to counterbalance the number of ideas developed (quantity) in engineering design research because increases in the fluency of ideas must also be proportional to increases in the spread of the ideas [258].

Without exploration, designers may misconstrue the solution space to be very narrow [86]. One of the main contributing factors to this trend is functional fixation, or blind adherence to solutions that are familiar and comfortable, which can generally lead to products of lower quality or innovation [137, 147]. As such, it is not surprising that research in engineering design has shown a correlation between the amount of design space explored and the quality of the final design [90].

Measurement of design space explored requires measuring mathematical functions on groups of ideas [199]. To address the desire to measure the extent to which tools promote variety, Vargas-Hernandez *et al.* [235] developed the SVS metric with the intent to provide a repeatable and reliable method to calculate design variety by rewarding ideas that are differentiated at higher levels of abstraction. In SVS metric, the authors decompose design variety into four hierarchical levels: the physical principle, followed by the working principle, embodiment, and detail. Specifically, they proposed that design variety should be calculated as shown below in equation 2.1.

$$V = \sum_{j=1}^{m} (f_j) \sum_{k=1}^{4} (S_k.B_k)/N \tag{2.1}$$

where V is the variety score, m is the number of functions solved by the design, f_j is

a weight assigned to the relative importance of function j, S_k is the score for hierarchical level k, B_k is the number of branches at hierarchical level k, and N is the total number of ideas in the set. The key intuition behind this metric is that each idea is represented by hierarchical functions or attributes. Attributes on top of the hierarchy are more important than ones below, and if a set has multiple ideas with unique higher level attributes, then that set gets a higher variety score.

SVS metric has been criticized for double counting ideas at each level in the tree and for the selection of the weights at each level of the tree [193, 276]. Because of these pitfalls, Nelson *et al.* [193] refined the metric by seeking to account for the double counting of ideas present in the SVS metric by considering the number of differentiation at each hierarchical level rather than considering all the levels. In addition, Nelson *et al.* modified the SVS metric by altering the weighting scheme from 10, 6, 3 & 1 to 10, 5, 2 & 1 for the physical principle, the working principle, the embodiment, and detail respectively for their NM metric. They argued that the new weighting scheme assures that at least two ideas at a lower hierarchical level must be added to equal the variety gain by adding a single idea at the next higher hierarchical level [193]. However, both SVS and NM do not provide a definition for each level of the hierarchy. There have been insufficient justifications for weights used in genealogical tree metrics [175] which can lead to large variations in the deployment of the metric in engineering design research. Other improvements of SVS metric includes the work of Verhaegen *et al.* [268], who combined Shah's metric with a Herfindahl index based tree entropy penalty, to encourage "uniformness of distribution" — essentially preferring trees that have even branching. Outside design, researchers have measured the breadth of ideation using metrics like mean pairwise distance

between ideas [55] or by manually subgrouping functions into categories [56].

The new metric proposed in our work is closest in scope to Fuge *et al.* [99], who showed that both SVS and Verhaegen's metric were instances of submodular functions and argued that variety metrics are coverage functions which should belong to this family of functions. They introduced a probabilistic model that computes a family of repeatable variety metrics trained on expert data. In the Research Task 2, we propose a new metric based on the Herfindahl index, which does not necessitate finding hierarchical features. Our metric also satisfies the properties of supermodularity (function whose negative is a submodular function), which allows us to optimize variety using a greedy heuristic algorithm. We show that unlike past metrics, this new metric has better alignment with the judgment of variety by people.

2.3 Research Gaps and Research Objectives

To the best of our knowledge, the literature reviewed above displays gaps in novelty and variety computation where the existing metrics often (if not always) lack in repeatability, validity and explainability. For novelty computation, current subjective novelty metrics like CAT often act as a black box, with no avenue to understand the criteria used by experts. They are often not repeatable too as they require an expert to remember all designs in the domain (sometimes thousands) and assign scores on an absolute scale. To address these issues in the first research task, we propose a new novelty computation method where we show how two-dimensional maps can provide insights into novelty computation and provide a path to represent subjective opinions as numerical values. In

our method, we ask subjective triplet queries from raters and estimate a low dimensional embedding for design items. We use these idea maps to estimate the novelty score as well as provide insights into the decision making of raters.

We also found gaps in literature regarding the lack of validity and sensitivity in existing variety metrics. Commonly used variety metrics SVS and NM are often not able to distinguish between different sets of designs and may not align with how humans perceive variety. Our experiments validate these gaps. In the second research task, we propose a new variety metric and show that it is more accurate and sensitive than existing metrics. We also provide a methodology to test any new variety metric against human feedback.

2.4 Research Task 1: Data-driven Novelty Metrics

In this work, we focus on learning idea maps or design embeddings, *i.e.*, an embedding in which similar ideas lie close together and dissimilar ideas are far apart, entirely based on the similarity-triplets supervision provided by a person. We show how studying idea maps allows us to understand what factors may be important for different individuals in judging similarity and how these embeddings can be used to rate ideas on novelty. The next section provides an overview of the methodology used, followed by our experimental results on two design domains. We discuss the limitations and design implications, followed by a discussion on the extension of this method to study design novelty.

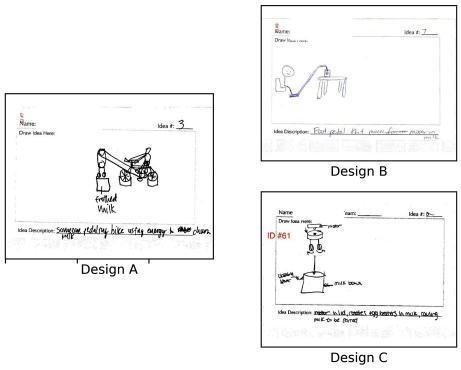

Figure 2.2: Example of a triplet query asked from the raters in our experiment. A rater answers the question: "Which design is more similar to design A?"

2.4.1 Methodology

Below, we discuss how triplet responses can be used to estimate idea maps and define two novelty metrics based on these maps. The process is outlined in Fig. 2.3.

2.4.1.1 Idea Map Generation

Given a set of N designs, we first generate all possible triplet queries from them. This set of queries is given to raters as surveys. After collecting responses, we use the Generalized Non-metric Multidimensional Scaling (GNMDS) technique [5] to find embeddings of design ideas.[2] The idea map obtained by applying GNMDS to the triplet responses by a rater tries to satisfy a majority of the triplets. To do so, GNMDS finds a low-rank kernel matrix K in such a way that the pairwise distances between the embedding of the

[2]Before selecting GNMDS, we compared it to three other common techniques — Crowd Kernel Learning, Stochastic Triplet Embedding and t-Distributed Stochastic Triplet Embedding — for our data. We did not find major differences in the percentage of triplets satisfied between different methods.

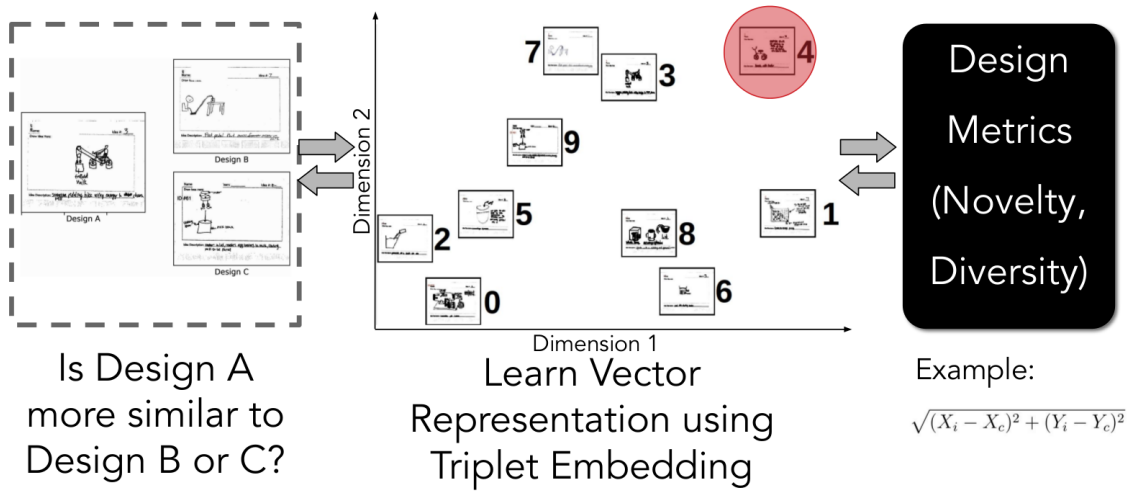

Figure 2.3: Process map to calculate design metrics using triplet embeddings.

objects in the Reproducing Kernel Hilbert Space (RKHS) satisfy the triplet constraints with a large margin. It minimizes the trace-norm of the kernel in order to approximately minimize its rank, which leads to a convex minimization problem. Figure 2.2 shows an example of a triplet query with three design sketches used in our study. We represent the response of the rater to any query as 'ABC' or 'ACB'. Response coded as 'ABC' means Design A is closer to Design B than Design C and response coded as 'ACB' means Design A is closer to Design C than Design B. GNMDS method allows the triplets to contradict; this can often happen when multiple people vote and use different criteria in finding item similarity. The resulting output is x,y coordinates for each design item.

2.4.1.2 Measuring Novelty on a Map

Given an idea map, our goal is to calculate the novelty score of each idea. As nearby ideas on the map denote similarity with each other, one would expect that the idea furthest away from everyone else will also be the most novel within the set. As the novelty of an item

in a set can be interpreted as how unique or dissimilar an item is [267], the problem is equivalent to finding ideas which are distant from all other ideas on the map. However, many different methods exist to find outliers on a two dimensional map. Here, we define two such metrics which give a high score to ideas which are away from everyone else on a map. We name these metrics as Nov_{sum} and Nov_{cent}, which score any item i as follows:

$$Nov_{sum}\ (i) = \sum_{j=1}^{N} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \tag{2.2}$$

$$Nov_{cent}\ (i) = \sqrt{(X_i - X_c)^2 + (Y_i - Y_c)^2} \tag{2.3}$$

Here X_i, Y_i are the 2-D coordinates of idea i. X_c is the 2-D coordinates of the centroid of all ideas. Nov_{sum} defines novelty of an idea in a set as the sum of distances from the idea to all other ideas. This simple formulation has been used in the past for document summarization to define representative items [171]. It assumes that the most novel idea has the highest average distance from all other ideas. Nov_{cent} defines the novelty of an idea as the distance from the centroid and has been used in [183] to measure novelty. The centroid is a theoretical point in the space, created by averaging the attributed values across all designs in the space. It assumes that the most novel idea is the idea which is furthest away from the centroid of all the ideas. By giving a high score to ideas furthest away, both metrics allow us to rank order all ideas.

We experimented with a few other methods to measure novelty of items on a map but chose these two metrics as they are a) easy to compute and b) make few assumptions about the distribution of ideas or how ideas are clustered in the map. Note that it is possible to discuss many more metrics for novelty detection on a two-dimensional map, however, we only use these two metrics to show how triplet embeddings enable novelty

calculation. We did not aim at finding the best novelty metric for any given domain as it will depend on how one defines the qualities needed for a metric to be called 'best' for a particular domain. It is unlikely that one novelty metric generalizes to many domains.

2.4.1.3 Measuring Rater Performance

Triplet responses given by raters can vary in accuracy or reliability due to factors like rater expertise or motivation. However, it is difficult to assess the quality of triplets by measuring intra-rater reliability, as they are a subjective assessment of how a rater views the similarity of ideas. Instead, we estimate a rater's performance by measuring how consistent they are with their own responses using two methods. First, we estimate the self consistency of raters by adding additional triplet queries, which are repeats of existing queries. Second, we measure the number of violations a rater makes in the transitive property of inequality; for example, suppose a rater gives two responses as ABC and CAB, which means that she finds item A more similar to item B and item C more similar to item A. These responses imply $AB < AC$ and $CA < CB$, where AB indicates how similar item A is to item B, AC indicates how similar item A is to item C and so on. These two inequalities imply the third inequality, that idea B is more similar to idea A ($BA < BC$). If this rater provides a third triplet response of BCA indicating idea B is more similar to idea C, then this violates transitive property — any two triplets are consistent, but not all three, so there is one violation of the transitive property.

We count the total number of transitive violations and the percentage of self-consistent answers as measures of rater performance.

2.4.1.4 Measuring Map Similarity

To find the similarity between two idea maps, we employ three different methods by comparing: a) 2-D positions, b) Distance vectors, c) The overlap between triplets obtained from each map.

To compare the 2-D position of points on two maps, they should be on the same scale. However, maps obtained by triplets or drawn by people can be on different scales and maybe rotated or translated. To overcome this problem, we use Procrustes analysis [107] to find the optimal scaling/dilation, rotations, and reflections such that the sum of the squares of the pointwise differences between the two input datasets is minimized. We call the least squared error after transformation of one map as the 'Disparity' score between the two sets of points.[3] However, this measure is dependent on an intermediary step of correctly solving another optimization problem, which may introduce error (if the Procrustes transformation converges to a local minimum).

To get more confidence in comparing two maps, we define two more map similarity methods. In the distance based method, we calculate the Euclidean distance vector of each point with every other point. For 10 points, we get 45 unique distances. We find the mean squared error (MSE) between the distance vectors of the two maps. Distances are rotation and translation invariant. We divide each distance vector by the maximum distance of that vector to make them scale invariant too. This resolves the issue of different map scaling by bounding the maximum distance for each map to one unit.

[3]Score calculated using Python scipy library: `https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.spatial.procrustes.html`

The above distance method gives a measure of how metric distances between the two maps differ. However, as the maps are generated using non-metric triplets, maps with different spatial arrangements can still satisfy the same set of triplets. Hence, we propose a new non-metric similarity measure between two maps called "Triplet error". In this method, we generate a set of triplet responses corresponding to each map such that it satisfies the given map exactly. Let us call these sets S_1 and S_2. This set of triplet response can be different from the triplet set from which the map is generated (as we will see in our experimental results, maps may not satisfy a small proportion of triplets provided by raters). We count the number of triplet responses which are common between the two maps. Triplet error is defined as the percentage overlap between these two sets of triplets *i.e.* $\frac{|S_1 \cap S_2|}{|S_1|}$. Triplet error measures how the two maps compare in relative distances of items.

To explain triplet error, we take an example of comparing two maps with four items each as shown in Fig. 2.4 b). Visually the two maps look different. However, if we list the triplet responses which satisfy the map on the left side, we get the following set of twelve triplets: ABC, ABD, ACD, BAD, BAC, BDC, CAB, CDA, CDB, DBA, DCA and DBC. As mentioned before, item ABC means A is closer to B than C. If we list the triplets satisfying the map on the right side, we get the exact same set of triplet responses. Hence, the triplet error is zero between these two maps. This measure is independent of scaling and allows similar maps to have different spatial arrangements. In comparing maps in our experiment section, we report all three measures.

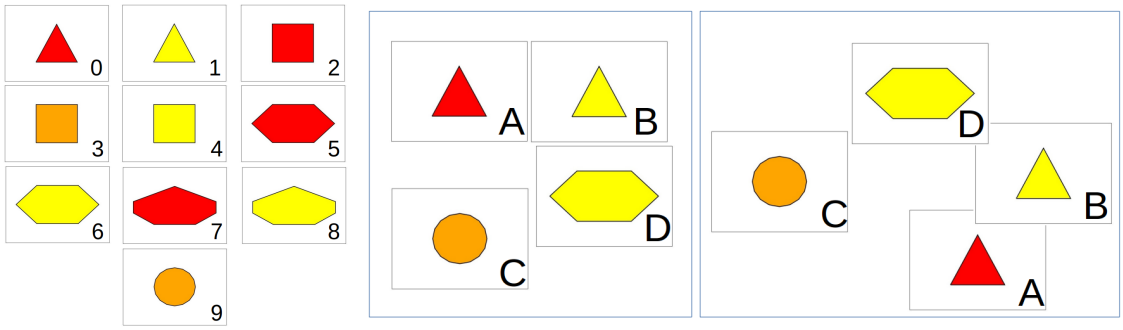

Figure 2.4: a) Dataset of ten polygons used in the first experiment. b) Two idea maps with four items each. Although these maps look different, they satisfy the same set of triplet queries.

2.4.2 Results

To demonstrate our methodology, we consider two case studies. We chose the first case study, such that the idea maps generated are simple to understand and the novelty measure is easily verifiable. By selecting items with only a few attributes, we can estimate the ground truth of novelty estimation. In contrast, for the second case study, we select a complex design domain, where "ground truth" is not known and different raters may disagree on what defines being novel. With this guiding principle, in the first study, we generate a dataset with ten colored polygons, who are rated by eleven raters. We show two dimensional idea maps and novel items discovered for different raters in a seemingly simple design domain. In the second study, we selected ten milk-frother sketches from a real-world ideation exercise conducted in [242]. Here we show how individuals vary in defining similarities between complex designs and how their ratings can be aggregated to generate meaningful idea maps. We also ask raters to generate physical maps directly and compare them to idea maps obtained using embeddings.

2.4.2.1 Experiment 1: Colored Polygons

Our dataset of ten polygons is shown in Fig. 2.4 a), which contains two triangles, three squares, two hexagons, two heptagons and one circle. We obtain 360 triplet queries (all possible permutations of three items) from these ten sketches and show them to eleven raters. The raters comprised one Ph.D. student (Industrial Engineering), one Master's student (Mechanical Engineering) and nine under-graduates (Psychology). Suppose a given triplet has items A, B and C as polygons 7, 6 and 2 from Fig. 2.4 a) respectively. For this triplet, raters have to decide whether they find the red heptagon more similar to the yellow hexagon or the red square. One rater may prioritize color-based similarity to shape and thus answer "the red heptagon is more similar to the red square," while another may use closeness in area of polygons to answer "the red heptagon is more similar to the yellow hexagon". To gain insights into their decision making process, we also ask raters to explain their choice for 20 randomly selected triplets. These responses helped verify our hypotheses about the factors considered by each rater.

Automated Rater: To verify that the triplet generated maps correctly reflect provided triplet responses, we first use an automated rater who rates all triplet queries consistently based on a fixed set of rules. We define the rules such that this automated rater always rates polygons with the least difference in the number of sides as more similar. When two polygons B and C have similar priority in the previous rule, it selects the polygon which is more similar in color to base polygon A. As the automated rater uses consistent rules for all triplets, we find that its self consistency score is 100% and it has zero

transitive violations, as expected. The resultant idea map obtained from the automated raters triplet ratings is shown in Fig. 2.5. One can notice from this idea map that similarly shaped items are grouped together. As one might expect, the two dimensions that can be identified from this idea map are color and shape. Polygons of similar shape are grouped together, while yellow colored polygons are placed slightly below their red counterparts. The gap between triangles and squares is lesser compared to the gap between squares and hexagons. This is because triplets with less difference in their number of sides are rated as more similar by the automated rater. Hence, this map can be considered a good representation of the triplet ratings provided by the automated rater.

In contrast to the automated rater, human raters may not always use consistent rules. Different people may give different priority to polygon attributes like color, shape, symmetry *etc.* We summarize our results for 11 raters in Table 2.1. Column 2 lists the self consistency score for each rater and column 5 lists the count of transitive violations. Column 3 and 4 provide the top 3 items calculated using the two novelty metrics discussed before. Column 6 reports the percentage of triplet responses not satisfied by each map found using embedding method (lower is better).

Let us take the example of idea maps obtained for two raters (rater ID 5 and rater ID 9 from Table 2.1 respectively). Idea map of Rater 5, shown in Fig. 2.6 places similarly shaped polygons near to each other. We also notice that red colored polygons are placed above yellow ones, similar to the automated rater. This provides evidence that this rater used the shape and color as the main criteria to answer triplet queries. In contrast, the placement of similarly colored items together for the map of Rater 9 (Fig. 2.7) indicates that color is more important to her than shape. The orange square is closer to the orange

circle in her map and far from similarly shaped squares.

When we look at the explanation provided by Rater 5 for a subset of queries, she repeatedly mentions "My choice was made by determining which polygon had a number of sides closest to polygon A" while Rater 9 mentions many of her triplet comparisons were decided based on "color, shape, number of sides". Hence, the criteria used by individual raters are reflected in their idea maps, grouping similarly colored or shaped items together.

Given the idea maps of these ten polygons, one would expect the most novel item to be most dissimilar to all other polygons. For Rater 5, Figure 2.6 shows that the circle is far away from all other polygons and thus one may consider it novel with respect to other polygons present in the dataset.

Table 2.1 shows the top three most novel sketches for each rater using the two novelty metrics. We find that the two metrics give the same set of top three items for 9 raters and the remaining 4 have atleast 2 items common. This shows that the two metrics align in their novelty assessment. We also find that the orange circle (Polygon 9) appears in top three for most raters, indicating novelty metrics indicate the circle as the most novel item and there is a consensus among raters that it is the most novel item in the set. This matches our expectations, as we designed this dataset such that the circle is of a different color and unique shape compared to all other items in the set. The main takeaway from this experiment is that by studying individual idea maps and calculating novelty measure of items on these maps, we can calculate the most novel items as well as understand the factors which individuals consider in deciding item similarity.

Rater ID	Self consistency (%)	Top three Nov_{sum}	Top three Nov_{cent}	Transitive violations	Triplets not satisfied (%)
AR	100.0	9, 1, 0	9, 1, 0	0	5
1	83.3	9, 1, 0	9, 1, 0	2	11
2	100.0	9, 0, 8	9, 8, 0	3	11
3	83.3	9, 4, 8	9, 8, 4	3	15
4	75.0	9, 1, 0	9, 1, 8	2	15
5	100.0	9, 1, 0	9, 1, 0	0	1
6	100.0	9, 1, 0	9, 1, 0	0	15
7	91.6	9, 1, 0	9, 1, 0	0	15
8	91.6	9, 1, 6	9, 1, 6	8	21
9	83.3	1, 9, 0	1, 9, 0	9	22
10	83.3	9, 1, 3	9, 1, 0	4	10
11	100.0	9, 8, 1	9, 8, 6	0	15

Table 2.1: Rater performance and top three novel items for different raters of experiment on polygons. We find that most raters find the circle (item 9) as the most novel polygon.

Rater ID	Self consistency (%)	Top three Nov_{sum}	Top three Nov_{cent}	Transitive violations	Triplets not satisfied (%)
1	91.6	5, 2, 4	5, 2, 4	5	17
2	50.0	6, 0, 2	6, 0, 2	5	21
3	83.3	1, 2, 7	1, 7, 0	5	20
4	75.0	4, 0, 6	0, 4, 6	10	20
5	75.0	2, 8, 5	2, 8, 3	10	21
6	58.3	1, 4, 5	1, 4, 5	20	27
7	41.6	4, 1, 2	4, 2, 1	8	15
8	41.6	1, 7, 4	1, 4, 7	20	26
9	58.3	0, 6, 1	0, 6, 2	11	16
10	75.0	4, 0, 1	4, 0, 2	12	19
11	58.3%	5, 6, 2	5, 0, 6	5	16

Table 2.2: Rater performance and top three novel items for different raters of experiment on design sketches.

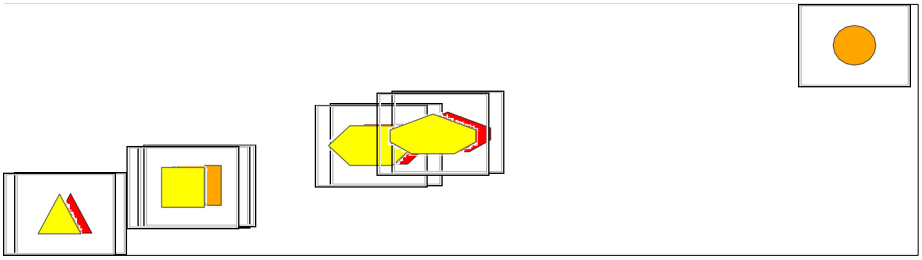

Figure 2.5: A two-dimensional embedding for the automated rater of polygons example.

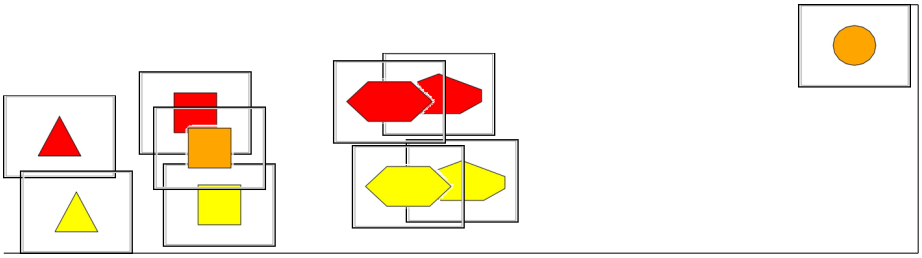

Figure 2.6: A two-dimensional embedding obtained from ratings by Rater 5 *on the polygon example. From the embedding, we notice that Rater* 5 *uses the number of sides as the primary criteria for her triplet decisions.*

2.4.2.2 Experiment 2: Design Sketches

In this experiment, we find the embeddings for ten design sketches of milk-frothers. This set of design sketches is adopted from a larger dataset of milk-frother sketches [242, 256]. To create the original dataset, the authors recruited engineering students in the same first-year introduction to engineering design course. The task provided to the students was as follows: *"Your task is to develop concepts for a new, innovative, product that can froth milk in a short amount of time. This product should be able to be used by the consumer with minimal instruction. Focus on developing ideas relating to both the form*

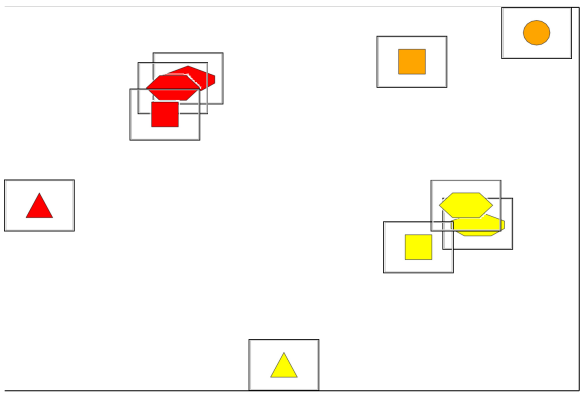

Figure 2.7: A two-dimensional embedding obtained from ratings by Rater 9 on the polygon example. Rater 9 self reported that she used color, shape, and the number of sides as key factors in answering triplet queries.

and function of the product". Details of experiment to collect data are available online.[4]

We selected ten design sketches from this dataset for this experiment. Fig. 2.8 shows these sketches. As shrinking the sketches and their overlap makes it difficult to understand a 2-D map, we allocate number ids to each sketch and plot the numbers on idea maps instead. Similar to the previous case, eleven raters were used in this experiment. The raters comprised of one professor (Industrial Engineering), two Ph.D. students (Industrial Engineering) and seven undergraduate students (Psychology).

Figure 2.9 a) and 2.9 b) show the idea maps obtained by Rater 7 and Rater 10. These maps provide useful cues into the decision-making process of these raters, who used different decision-making criteria. The embedding of Rater 7 in Fig. 2.9 a) provides evidence that she might have grouped sketches which have cup to store milk in the design as more similar (as shown by sketches 6, 5, 2 and 7). She also grouped sketches 4 and

[4] `http://www.engr.psu.edu/britelab/resources.html`

3 nearby, both of which have bikes in the design. Similarly, Rater 10 also has sketches 4 and 3 nearby but 6, 5, 2 and 7 are not nearby. To understand the rationale used by the two raters, we qualitatively analyzed their explanations. For the triplet query shown in Fig. 2.2, Rater 7 finds sketch C as more similar to sketch A and mentions her choice as being based on "Simple or complex" design. Rater 10 finds sketch B as more similar to sketch A and gives the reason "it both spins and is powered by a person." We find Rater 7 mentions for many other triplet queries that she used design complexity as the primary criteria for judging which ideas are similar. She also gives the reason: "If it spins, or if it includes cups" for a few triplets, indicating that the presence of cup is an important criteria in her decision making.

In contrast, Rater 10, mentions a multitude of factors for different triplets like the method by which the milk was frothed (e.g. shaking), the form of the frother, if design had a motor, if something is being put into the milk or if the milk goes into something, *etc.* Due to the multitude of factors used by Rater 10, ideas in her map are possibly grouped due to a combination of different factors.

To verify the novelty calculation for Rater 10, we asked her to provide us a rank-ordered list of the most novel milk-frother sketches from this dataset. Her top three most novel sketches were $0, 1$ & 6. Nov_{sum} metric finds sketches $4, 0$ & 1 as the top three ideas from her idea map while Nov_{cent} finds $4, 0$ & 2 as the top three items. While the rankings don't completely overlap, it should be noted that her top three sketches $(0, 1, 6)$ occur on the periphery of her idea map, showing that they are generally far away from other sketches. To further compare our results with existing methods, we also coded different attributes for all 10 sketches and use the SVS novelty metric (abbreviated as

SVS_n) to calculate their novelty scores. The scores are: 0.718, 0.585, 0.6, 0.692, 0.566, 0.483, 0.585, 0.612, 0.45 and 0.715 for sketches 1 to 10 respectively. Using SVS scores, we find Sketches $0, 9$ & 3 are most novel, which is different from the subjective ranking provided by the rater as well as the scores calculated using idea maps. The difference can be attributed to factors considered in SVS_n score calculation.

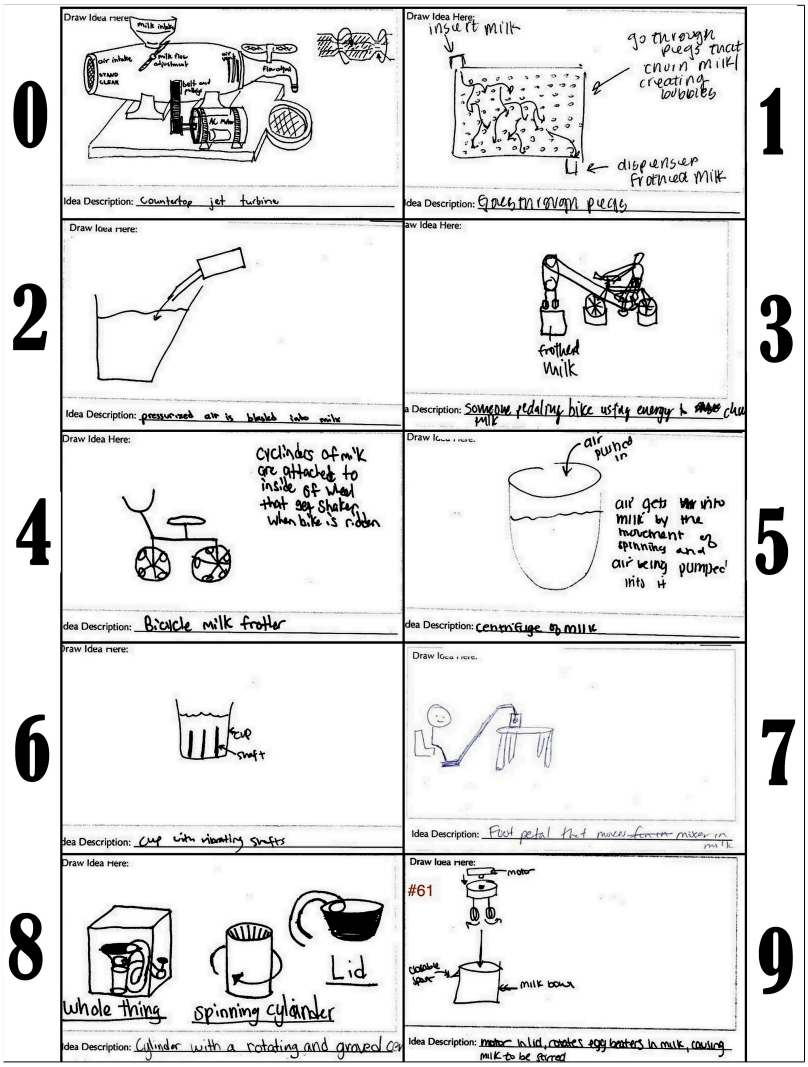

Figure 2.8: Ten milk-frother sketches used in the second experiment.

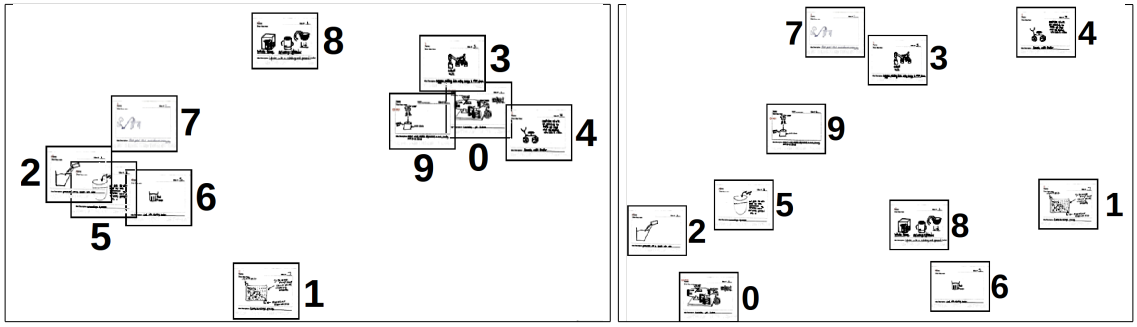

Figure 2.9: a) Idea map of design sketches for Rater 7*. Center of the sketch represents the 2-D position of embedding. Two main clusters can be seen. b) Idea map of design sketches for Rater* 10*. Center of the sketch represents the 2-D position of embedding.*

Wisdom of the Crowd: Table 2.2 shows the self-consistency score, transitive violations and top three most novel sketches for all users. As expected, maps of different raters differed from each other, which led to most novel ideas calculated using Eq. 2.2 differing too. As one would expect, we noticed that self-consistency scores and transitive violations are larger for design sketches compared to polygons experiment, implying that it is more difficult to judge real-world sketches compared to polygons.

To understand how sketches are grouped together, we combine the triplet responses of all raters and obtain a joint idea map. Fig. 2.10 shows the joint map of all eleven raters. As we add all triplets from raters who considered different (unknown) factors in judging idea similarity, the aggregated map can be considered to represent an average of all such attributes. One can study this map to find meaningful clusters in it and see which ideas are grouped together. For instance, on the right-hand side, we see three sketches (sketch 2, 5 and 6) clustered together, each of which uses a cup to hold milk. On the left-hand side, we see two sketches with bikes (sketch 3 and 4) clustered together. Two complex designs

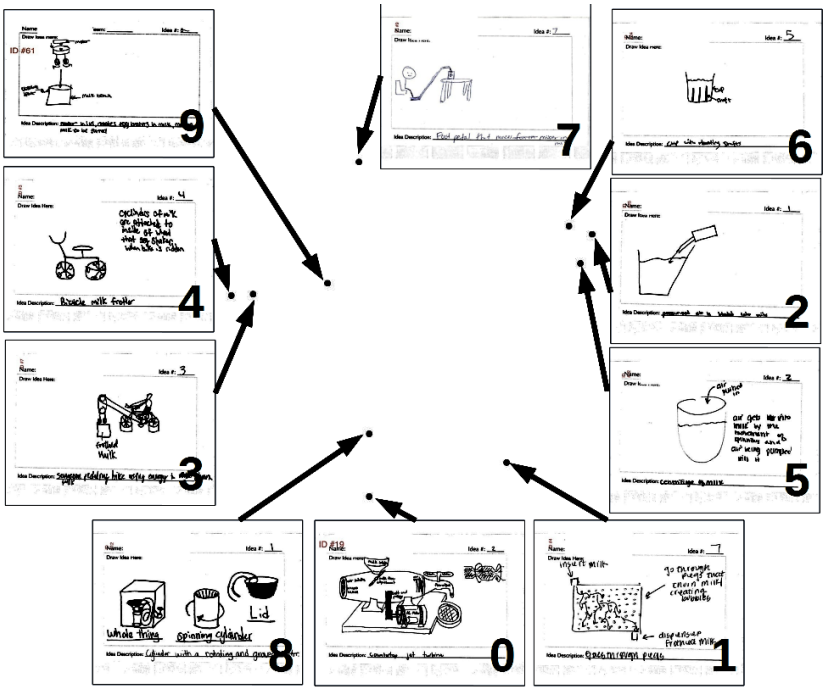

Figure 2.10: An idea map obtained by combining triplets from all raters and using triplet embedding method. The ID of each sketch is shown at the bottom right corner.

(sketch 0 and 8) with multiple moving parts are clustered together at the bottom. Using this map and our novelty metric, we find the most novel idea is Sketch 0, while the least novel is sketch 9. Sketch 0 is at the bottom of the map in Fig. 2.10, distant from all other sketches. As noted before, sketch 0 proposing a counter-top jet turbine to froth milk is the most novel sketch rated by the expert too. While individual idea maps of different raters disagreed on scoring the most novel sketch (due to different criteria used), we also found that sketch 9 ranked among the least novel items by majority of the raters.

So far, we have shown how individual idea maps can provide cues into factors important for raters in judging idea similarity. We have also shown how a joint map of multiple raters meaningfully groups sketches and can be used to estimate explainable

novelty of sketches. Next, we measure how the raters differ from each other in their triplet responses.

Similarity between Raters: To compare the similarity between triplet responses of different raters, we represented their responses as a one-hot encoded binary vector of length 720 and found cosine similarity between these vectors.

We applied multiple clustering methods to identify groups among these users and identified two clusters. We found that raters 1, 3, 5 and 10 are in first cluster and all other raters are in second cluster. Interestingly, Rater 5 and Rater 10 were the two experts in our rater pool and we found that they were also clustered together, along with Rater 1 and Rater 3. We then calculated the similarity matrices for each user's idea map and found the matrix distance between different idea maps. We again clustered the raters using the distance between their maps and found that they likewise group into two clusters. This finding is important, as we are able to find two supposedly non-experts, who are indistinguishable from experts based on their triplet ratings. Such groupings can be used to find aggregated maps for each group and study differences between idea maps of a group of raters.

Sketches that are Difficult to Judge: Different sketches have different levels of complexity. Some sketches in a triplet query can be considered similar/dissimilar based on multiple factors due to their design complexity (like sketch 0) but others may be simple in design and judged on fewer factors (like sketch 2). Finding sketches that are consistently difficult to judge by raters is important, as it can help understand features within these difficult sketches which cause disagreement among raters. To understand which sketches

are more ambiguous or are difficult to rate, we measure the total number of times a sketch appears in triplets where raters disagreed. For instance, if 50% of raters give Design B as triplet response and other 50% give Design C, then all three sketches in this triplet are considered difficult to rate. We measure disagreement by the Shannon entropy of all responses and we calculate the score of each sketch by adding the entropy from all triplets for all raters in which it appears. Using this score, we find that sketch 8 has the highest disagreement score among raters, followed by sketch 0. Sketch 1 followed by sketch 6 have the least disagreement scores. This indicates whenever sketch 8 appeared in a triplet, raters were more likely to give different responses. One possible reason for this can be design complexity. Sketch 8 and sketch 0 have many moving parts and are more detailed sketches, hence they can be interpreted differently by different raters compared to some other sketches which are simpler in design.

In the next section, we show that the embedding obtained by combining the triplets of multiple raters is robust. We show this using two experiments. First, we reduce the number of triplets available to derive the embedding and show that we can obtain a similar map using only a small fraction of triplet ratings originally used. Second, we add noise to the triplet ratings by flipping a percentage of triplets (simulating mistakes by raters) and show that these maps are resilient to significant levels of noise too.

Maps Using Fewer Triplets: As mentioned before, we collected 360 similarity judgments each from 11 raters for both experiments. This task is time-consuming and difficult to scale as the number of sketches grows. However, past researchers have found that one can obtain a meaningful embedding with fewer triplets [26]. To empirically measure how

many triplets are needed to obtain an embedding close to the one obtained in Fig. 2.10, we varied the number of triplet ratings available to us and found different embeddings. As different embeddings cannot be directly compared, we calculate the triplet error of each embedding with baseline embedding of Fig. 2.10. For any given percentage of triplets to be used, we performed 100 runs with different subsets. Figure 2.11 a) shows the resultant median triplet error along with 5^{th} and 95^{th} percentile. We found that using a small fraction of, say, 30% of available triplets, the median triplet error is only 9.1%. Hence, one can significantly reduce the number of triplets needed to find these embeddings. Our approach can be combined with active learning approaches to minimize the number of triplet queries needed to construct meaningful embeddings for larger datasets.

Maps Using Noisy Triplets: In Table 2.2, we notice that a few raters have low self-consistency scores and suffer from multiple transitive violations. To study how such noise can affect the idea map, we conduct an experiment to simulate noisy responses. We use all the 3960 triplet queries obtained from 11 raters, but randomly flip the response for a percentage of those triplets. This situation can occur in cases where rater accuracy goes down due to fatigue, when a few raters intentionally lie about similarity judgments, rater changes the criteria to judge similarities while doing the survey, human error, *etc.* To measure the effect of noise, we assume the map shown in Fig. 2.11 b) as the ground truth and compare it to maps obtained from noisy labels using the triplet error metric. Figure 2.11 b) shows the variation of the triplet error from the baseline idea map (Fig. 2.10) with increasing noise percentage. When 25% of triplets are flipped, the median triplet error is only 8.3%. To understand how much triplet error is acceptable, we refer the readers to the

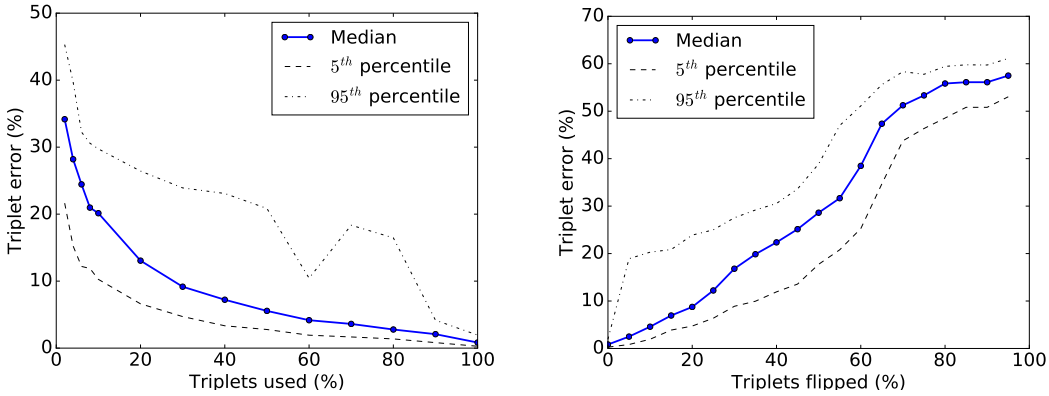

Figure 2.11: a) Triplet error between idea maps of embedding shown in Fig. 2.10 and embedding obtained using a subset of triplet ratings. We use 100 runs with different subsets of data to obtain the embeddings. Using only 30% of the total triplet responses, we find that the median error is less than 10%. This shows that triplets often provide redundant information. b) Triplet error between embedding generated using noisy triplets compared to embedding shown in Fig. 2.10. We perform 100 runs and flip a subset of triplets randomly to obtain the embeddings. A small increase in the median error shows that idea maps are robust to a small percentage of false ratings by raters. Even if people are inaccurate in a small percentage of their responses, the maps do not change much.

comparison of a physical map with a triplet map in the next section. Here, triplet error of 18% can occur in reasonably similar maps with few items misaligned (Fig. 2.12). This shows that although increasing noise changes the idea map, this approach is still resilient to significant levels of noise.

2.4.2.3 Comparison with Human Generated Maps

So far we have generated and compared idea maps created using only the triplet responses. How do these algorithmically-generated maps compare to a map that the same rater would generate directly (*i.e.*, by placing ideas on a 2D surface) without the intermediate step of answering the triplets? In this section, we conduct an additional experiment to generate idea maps directly from raters and then compare these idea maps with the maps generated using embedding methods.

Participants: Four subjects were selected from the group of raters that had participated in the triplet surveys. The subjects were selected based on their consistency in answering the triplet surveys. The participants comprised of 1 Faculty member, 2 Doctoral students, and 1 Undergraduate student. The participants were given a six-month period between taking the survey and participating in the following experiment to avoid priming, and obtain results that are unbiased with respect to their original triplet responses.

Experimental Setup: Each participant was provided with the same 10 idea sketches utilized in the triplet survey, printed on 8.5" x 5.5" sheets of paper. The order of the ideas was randomized for each participant. The subjects were required to pin the sketches on a 65" x 55" canvas, such that the distance between any two sketches would be proportional to how similar they were to each other. The sketches were allowed to overlap. The subjects were allowed to move the sketches multiple times until they were satisfied with the idea map created. The participants were allotted a maximum time of 30 minutes for the activity. The participants were required to think aloud as they placed and moved the ideas

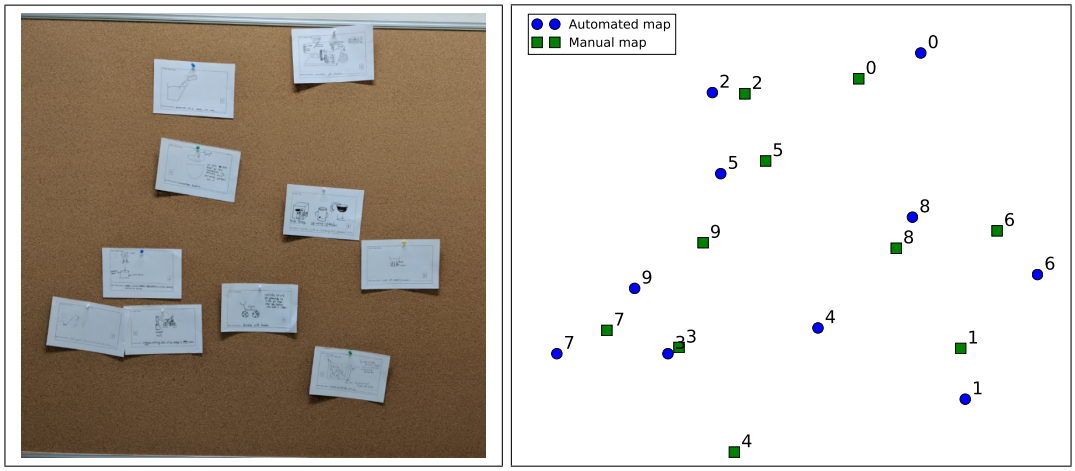

Figure 2.12: a) Photo of the map created by a participant by directly positioning idea sketches on a pin board. b) Correspondence between human generated map and the corresponding triplet map for Rater 10 *after correcting for scale and rotation. We notice that apart from sketch* 4*, most sketches have similar positions in both cases, showing that the participants idea map aligns with how she thinks about similarity between items.*

around on the canvas. Throughout the activity, the participants were recorded using audio and video equipment. Figure 2.12 a) shows how the maps were pinned on a cork board by one of the raters participating in this experiment.

Comparison between Automated and Manual Maps: We compare manual idea maps with automated idea maps (generated using triplet embedding methods), using the different metrics defined in the previous sections: 2-D position based, distance based and triplet error based.

Figure 2.12 b) shows the manual map and automated map overlaid on each other for Rater 10. We notice that her automated and manual map align well, as seen by similar numbers (sketch id) positioned nearby each other. Table 2.3 summarizes the results for

Rater ID	Percent triplet satisfied	Triplet error (%)	Disparity	Distance MSE
1	26.9	28.1	0.31	0.064
3	34.1	36.2	0.47	0.082
5	39.1	41.7	0.52	0.092
10	25.8	18.1	0.15	0.023

Table 2.3: Comparison between maps created manually by four raters and their automated triplet embedding maps. We observe from the low percentage of triplets satisfied that maps directly made by people are not great at satisfying their own triplet responses.

all four raters. Column 2 provides the percentage of triplets satisfied by the human-generated map. It measures the percentage of triplet inequalities (from the survey taken by the same rater) that are satisfied by the map generated by the person. Column 3 gives the triplet error between map obtained using an automated method and manual map. We notice that this error increases for raters who have low self-consistency (column 1). The disparity measure and mean squared error have a similar trend as triplet error. We notice that Rater 10 has the highest alignment between her manual and automated maps using all three metrics, while Rater 5 has the least alignment. The lack of alignment can be explained by a variety of factors like change in similarity criteria or lack of difficulty in creating manual maps which satisfy all preferences.

2.4.2.4 Design Implications

In this work, we propose using idea maps obtained from simple triplet queries to visualize design domain and measure idea novelty. Our experimental results have wide-ranging implications in many design applications as listed below:

1. Generating idea maps using triplet queries is not limited to sketches and can be used for other type of design artifacts like CAD models or text documents to assess human perceived similarity. For larger datasets, one can use a small sample of design ideas with triplet queries to understand features which are given more importance in defining similarity of ideas. These features can then be used to build feature trees for the entire dataset.

2. Generating such maps can help in understanding the design domain. For instance, one can use maps to understand what features are more important in defining similarity between ideas. We find in our experimental results that raters form identifiable clusters in idea maps. This could mean a whole new way of finding and studying fine-grained details in how people reason about concepts and designs. One can also measure changes in idea maps of a person or team before and after some trigger event (like showing analogies) to understand change in perception of design space.

3. In our experimental results, we found that humans, even experts, are surprisingly inconsistent. This measure of inconsistency provides some evidence that subjective novelty ratings may often be inaccurate. Our experiments provide evidence that if human raters are inconsistent in comparing similarity of sets of three ideas, then

this inconsistency may translate when they provide subjective novelty ratings too. The latter task essentially requires comparing an idea with all other ideas in the domain, which is strictly harder problem than comparing three items at a time.

4. As raters are often inconsistent in their responses, we also show that triplet embeddings are fairly robust and can handle large noise conditions. This makes our method well suited for many applications where ratings are noisy or ambiguous. In comparing embedding methods and novelty metrics, future studies can take into account robustness to noise too.

5. As shown when clustering raters, we can measure the similarity between raters from their triplet responses. This similarity measure can be used to find groups of similar raters. These groupings can be used to find aggregated maps for different groups and study differences between idea maps of a group of raters. For example, it can help to unpack differences in how experts rate items compared to novices, or how groups of experts from different fields might differ. Measuring differences between raters can help in training them too, by understanding what features someone is not paying attention to and providing appropriate intervention to increase inter-rater reliability. By following our study with qualitative questions, one can also understand how individuals come up with criteria to decide between triplets.

6. We provide a principled way of finding hard-to-judge concepts/designs. Finding these designs is important when assembling ground sets for things like verifying new metrics or the correct implementation of existing one. One can also allocate experts to rate hard-to-judge designs and use novices for easier designs.

7. Finally, finding accurate similarity representation paves the way for defining new families of variety and novelty metrics, which can help assess ideas. In this work, we have used simple novelty metrics like sum of distances on a map, but other measures can also be defined to quantitatively measure novelty. For instance, after obtaining an embedding, one can use kernel PCA [126] to estimate novelty. One can also use volume based coverage methods like Determinantal Point Processes (DPP) [14] to give high score to ideas which have highest marginal gain in coverage. Similarity representation for sketches allows us to use methods like diverse subset selection [15] — methods which traditionally need vector representation of design items.

2.4.2.5 Assumptions and Limitations

Before adopting this methodology, one should be aware of various assumptions and limitations. This work makes the following assumptions:

1. We assume that people use an internal ordering of idea similarity to consistently answer different triplet queries. In reality, we often notice that people are non self-consistent, which may introduce noise in the idea maps.

2. We assume that the importance given by a person to different factors in deciding similarity does not change with time.

3. We assume that design sketches can be represented using a low dimensional embedding. This low dimensional embedding explains the similarity relationships between sketches. However, it is possible that multiple embeddings exist, describing

different types of relationships.

4. We assume that novelty can be calculated using distances on the low dimensional embedding found from triplets.

5. We assume that lack of self-consistency and transitive violations do not substantially affect the low dimensional embedding coordinates.

Next, we list the main limitations of our work. First, we used two small datasets of ten items to demonstrate our results. In the naïve implementation, the number of triplets required for a complete ordering is proportional to the cube of the number of design items. This makes scaling the method to large datasets seem difficult. We show in our experimental studies that a complete triplet set may not be needed to obtain meaningful embedding.

Second, the non-metric nature of queries creates few problems. It is insufficient to simply satisfy the triplet constraints in the embedding through pairwise distances. It is possible to construct very different embeddings whilst satisfying the same percentage of the similarity triplets as shown in Fig. 2.4 b). We can choose between different maps by adding constraints or terms in the objective function of the optimization problem using further information from users. Further research can be done in ways to optimize the idea map by incorporating additional user preference information. Apart from multiple possible embeddings, measuring novelty using metric distances is difficult due to the non-metric nature of queries. This problem can be overcome using non-metric novelty estimation methods [155].

Third, we assume that design sketches exist on a 2-D embedding and novelty can be interpreted as distance from all other items on this embedding. We chose two-dimensions due to two reasons. First, the 2-D assumption is important for map interpretability. Second, we experimented by increasing the number of dimensions and did not find a large drop in percentage of triplets satisfied using three or four dimensions. However, this observation may not generalize to new domains and higher dimensional embedding may be needed to represent designs. There is also potential to extend the formulation of novelty we used. While current metrics are simple and straightforward, they may have some limitations which can be overcome by using other novelty detection methods like Unsupervised Outlier Detection using Local Outlier Factor (LOF) [41].

Fourth, pairwise comparisons can be affected by the independence of irrelevant alternatives (IIA) axiom [204]. In simple words, if a choice B is preferred to C out of the choice set B,C, introducing a third option D, expanding the choice set to B,C,D, must not make C preferable to B. This means that raters have an internal absolute scale, based on which they choose options. Experiments have shown that human behavior rarely adheres to this axiom [187]. We believe that if any idea set contains duplicate ideas, then the condition will not hold for triplet comparisons. This can be argued by a thought experiment. If Idea B and Idea C are equally preferable to a person, then introducing a new Idea D (which is a slight variant of Idea B) changes the probability of Idea B being selected by that person, violating IIA condition. However, as we use the triplet ratings only to find a low dimensional embedding (and not to rank order all ideas), the axiom should not affect our findings. Future research can investigate decision theory approach to study how IIA axiom affects triplet choices given by raters for design ideas.

Finally, different raters may use different criteria in deciding whether Item A is more similar to Item B or Item C. Ideas were only assessed by the raters at the idea level, not the feature level. Although, averaging the results of multiple raters provides a good estimate of aggregate view, the problem is inherently of multiple views. In future work, we will explore directly optimizing for multiple maps using multi-view triplet embeddings [25]. This will allow us to obtain multiple maps for each rater corresponding to different factors they considered.

2.4.3 Concluding Remarks of Research Task 1

In this task, we proposed a new way to derive novelty measure using low dimensional embedding of design ideas using subjective pairwise comparisons. Interpreting these idea maps gave insights into what items are considered similar by participants or a group of participants, what attributes are considered important by them in judging similarity and what design items are harder to judge. We show how these idea maps can be used to explain and measure novelty of ideas, where novelty of an idea is measured by how far it is from all other ideas on a map. We use two domains as examples — a set of polygons with known differentiation factors and a set of milk-frother sketchers whose factors are unknown. The validity of metrics is demonstrated by comparing them with people's ranking of novel items and by comparing maps generated against physical maps made by participants. These maps highlighted interesting properties of how participants chose to differentiate concepts and how to group participants by similarity. They pave the way to use computational methods to reveal what makes ideas novel and allow easy interpretation of results by visualizing ideas on a 2-D map. We compared our results using

both a completely automated method and using maps made directly by participants.

2.5 Research Task 2: Data-driven Variety Metrics

A 'variety' metric [5] is used to measure how well a set of ideas explore the design space. Hence, they are a type of coverage metric (metrics which measure the span of a set of items). We demonstrate that many existing variety metrics in design have low sensitivity and do not align with human perception of variety (have low validity). To demonstrate these issues and overcome the shortcomings of existing metrics, we propose a new design variety metric based on the Herfindahl index [125, 217] and demonstrate its effectiveness on two datasets. We also propose a practical procedure for comparing variety metrics via constructing ground truth datasets from pairwise comparisons by experts.

2.5.1 Methodology

In this section, we first describe a variety measurement method using the Herfindahl–Hirschman Index. Next, we show an example of variety calculation using the new metric. We show that the new metric can be optimized using a simple greedy algorithm to find sets of ideas with the highest variety.

[5] 'Variety' and 'diversity' refer to the same metric. We use 'diversity' in subsequent chapters in studying ranking and matching algorithms. However, we use the term 'variety' instead of 'diversity' in this chapter as it is more commonly accepted term within Engineering Design domain.

2.5.1.1 The Herfindahl–Hirschman Index for Variety

Over the last twenty years, economists have become increasingly interested whether diversity among multiple distinct population groups enhances or impedes a society's economic and social development. To quantify the economic impact of diversity, one must first create a proper index that captures how one society divides into various factions or parts.

Starting from the Gini index [103], Economists have used various diversity indices to evaluate the degree of social, economic, cultural, and other dissimilarities among people, regions, and countries. Initially used as an income inequality measure, the Gini index was re-interpreted by Simpson [239] as the inverse Hirschman–Herfindahl index. That index measured industry concentration and was also used by Greenberg [109] for the measurement of linguistic diversity. The value of the index measures the probability that two randomly chosen individuals in society belong to different groups.

This Herfindahl index (also known as Herfindahl–Hirschman Index, HHI, or sometimes HHI-score) [125, 217] measures a firm's size relative to the industry and indicates the amount of competition among firms. For a market with K firms, HHI is calculated by squaring the market share (MS_i) of all firms ($i \in \{1, \cdots, K\}$) in a market and then summing the squares, as follows:

$$HHI = \sum_{i=1}^{K} (MS_i)^2 \tag{2.4}$$

In this section, we propose a variant of HHI-score that can measure the variety of a set of designs, called Herfindahl–Hirschman Index for Design (HHID). To do so, we

assume that we are given a set of designs S. Each design within set S is divided into hierarchical levels like functional principle, working principle, embodiment, and detail (similarly to SVS and NM above). We then calculate the HHID index for each level separately for the entire set. For example, the HHID index for the functional principle level is given by:

$$HHID_F(S) = \frac{\sum_{i=1}^{N_f} |FP_i|^2}{N^2} \tag{2.5}$$

In this, $|FP_i|$ is the number of designs using functional principle i and N_f is the total number of functional principles. N is the total number of designs in the set S. $HHID_F(S)$ varies between $1/N$ to 1. $HHID_F(S)$ measures the probability that two randomly chosen ideas in a set have different functional principles. Similarly, we can define HHID for working principle, embodiment and details. We define the total HHID variety metric of a set of designs by taking the weighted sum of these four metrics as follows:

$$\begin{aligned} HHID(S) = w_1 \frac{\sum_{i=1}^{N_f} |FP_i|^2}{N^2} + w_2 \frac{\sum_{j=1}^{N_w} |WP_j|^2}{N^2} + \\ w_3 \frac{\sum_{k=1}^{N_e} |EM_k|^2}{N^2} + w_4 \frac{\sum_{l=1}^{N_d} |DE_l|^2}{N^2} \end{aligned} \tag{2.6}$$

Here, $HHID(S)$ is the total HHID score for a set of designs S and the weights determine how much importance is given to each type of principle. The weights w_1, w_2, w_3 and w_4 can be chosen such that the resultant value is always between 0 and 1 (for example, we can constrain the sum of weights to be 1). For instance, if all factors are equally important, then $w_1 = w_2 = w_3 = w_4 = 1/4$. $|WP_j|$ is the number of designs in the set using working principle j, $|EM_k|$ is the number of designs using embodiment k

and $|DE_l|$ is the number of designs using detail level l. N_f, N_w, N_e and N_d are the total number of functional, working, embodiment and detail principles.

The definition of HHID (S) is not a supermodular function[6] if it is normalized by N^2. However, the HHID metric defined by us in Eq. 2.6 is supermodular when it is not normalized by N^2, which allows polynomial time greedy optimization of the metric. This means that when a design is added to a larger set, the increase in HHID score is larger compared to the case when the same design is added to a smaller set of designs. This property can be exploited to find sets of maximum diversity using a greedy algorithm [195], which guarantees that the variety of the greedy search solution will be within 63.2% (or $1 - \frac{1}{e}$) of the variety of the optimal solution.

2.5.1.2 Calculating Variety of a Set

To demonstrate HHID calculation, we take the set of designs shown in Fig. 2.13 as an illustrative toy example.

In Fig. 2.13, for the set shown on top, there are eight polygons ($N = 8$). There are four items with a rectangular shape, three items with an oval shape and one triangular. There are five red colored polygons, two blue and one green. Three items have a solid fill, two have shaded and three are empty inside. Without loss of generality, for this example, we assume that color is the functional principle of a polygon, shape is the working principle and shading is the embodiment. We assume that all three lev-

[6]Submodular functions are functions defined over sets that are designed to model diminishing marginal utility, which is the mathematical property one needs to model diversity or variety [99]. Supermodular functions are functions whose negative is a submodular function.

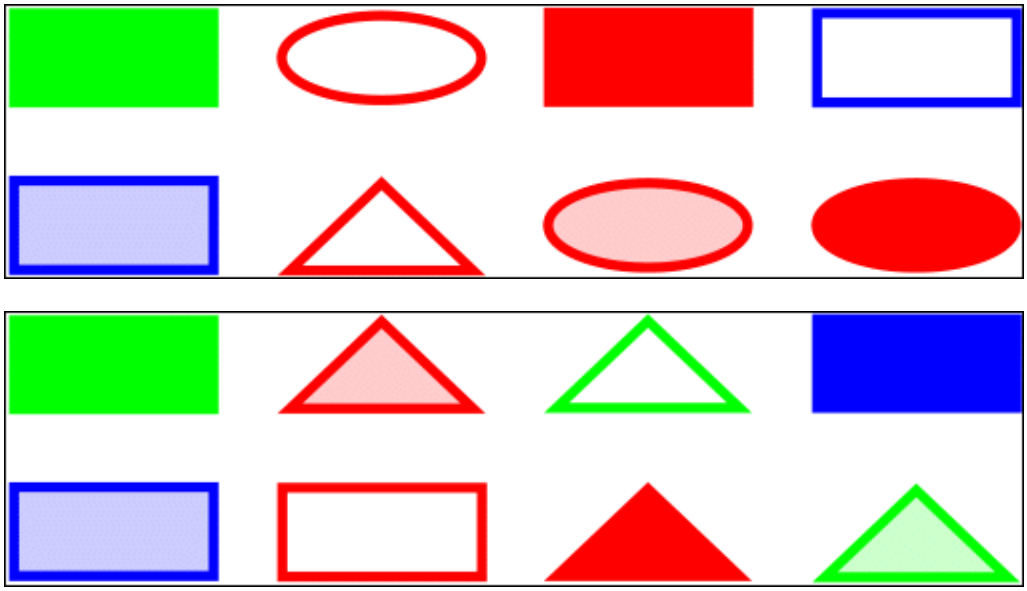

Figure 2.13: Example of two polygon sets (top shows Set A and bottom shows Set B) shown to participants in our experiment. Each participant answers the question: "Which set is more diverse?"

els are equally important in deciding the variety of Set A ($w_1 = w_2 = w_3$) and $N_f = 3$ as there are three unique functional principles (color). The $HHID_F$ score for color will be $(5/8)^2 + (2/8)^2 + (1/8)^2 = 0.47$. Similarly, $HHID_W$ score for shape will be $(4/8)^2 + (3/8)^2 + (1/8)^2 = 0.39$ and $HHID_E$ score for fill will be $(3/8)^2+(2/8)^2+(3/8)^2 = 0.34$. As we set all features to be equally important, then HHID for the set of designs is the average of the three numbers $(0.47 + 0.39 + 0.34)/3 = 0.40$. Similarly, the variety of any set of designs can be calculated.

2.5.1.3 Optimizing Variety of a Set

Using metrics like SVS, NM and HHID we can measure the variety of a given set of ideas (like the sets shown in Fig. 2.13). However, what happens when we want to choose the set

of eight polygons which have the maximum variety? One way is to enumerate all possible sets of size eight (about 2.2 million sets), calculate their variety score and then find the set with the highest variety score. This approach becomes intractable as the number of items in the ground set increases.

Another approach and the one we use is to leverage the mathematical properties of the variety function and find approximate solutions close to the optimal. To find sets of maximum variety, we use a sub-modular greedy algorithm (Algorithm 1) to order the ideas [195]. Given the set V of all ideas, the algorithm starts with an empty set $S = \{\}$ and add ideas to this set according to Algorithm 1. In the end, this set S will be the ranking that the algorithm outputs. It will contain all ideas ordered in such a way as to maximize the objective value defined in Eq. 2.6 (when the function is not normalized by N^2), *i.e.*, the ideas of high variety (*i.e.*, from principles less represented so far) are at the top of the ranking.

To achieve this, the algorithm starts adding ideas to an empty set S and removing them from set V, one idea at a time, such that the selected idea $i \in V$ is the one with the lowest marginal gain $\delta f(S \cup i)$ on set S. Here $\delta f(S \cup i) = HHID(S \cup i) - HHID(S)$. Here the set V is the set of all designs and set S is the selected set of design which we find using a greedy algorithm.

By choosing at each step to add the idea that will maximize variety (minimize the metric function) of the existing set of already added ideas, the algorithm not only selects the ideas but also orders them as well. Finally, as the function in Eq. 2.6 is super-modular and monotonic, the algorithm is also theoretically guaranteed to provide the best possible $(1 - \frac{1}{e})$ polynomial-time approximation to the optimal solution [94, 157].

Algorithm 1: Greedy algorithm used to obtain the maximum variety set. The algorithm performs a polynomial-time greedy maximization of the gain on the non-normalized HHID variety index. The output is a ranking of all ideas such that high-variety ideas are at the top.

Data: Original set V of all ideas

Result: Ranked set S of all ideas

initialization;

$S \leftarrow \emptyset$;

while $V \neq \emptyset$ **do**

    Pick an item V_i that maximizes $\delta f(S \cup i)$;

    $S = S \cup \{V_i\}$;

    $V = V - V_i$;

return S;

2.5.2 Results

We conducted two experiments to benchmark the proposed HHID metric with the commonly used SVS and NM metrics: (1) an experiment using a known and easily verifiable ground truth based on polygons, and (2) an experiment using design sketches provided by engineering students and rated by domain experts. Before introducing these experiments and their main results and implications, we describe how we constructed our experimental dataset of set comparisons for these two domains. As we have shown, constructing such sets is non-trivial, and one contribution of this work lies in describing a procedure for constructing such comparison sets for new domains.

2.5.2.1 Estimating Design Variety Ground Truth using Human Pairwise Comparisons

The first step in vetting design rating metrics is to identify a 'ground truth' of the measure that the metric is trying to capture and then calculate how accurate any given metric is in capturing that ground truth. However, for the case study presented here (design variety), ground truth estimation is difficult due to the large combinatorial space for sets of items and the lack of a benchmark dataset. For instance, a small set of thirty design ideas has more than one billion possible sets of designs for which variety can be calculated. Exhaustively calculating the ground truth is infeasible. Secondly, we do not use any existing variety metric to create the ground truth. Doing so would make the assumption that a given metric represents true variety, which is what the ground truth is used to

establish. Instead, we propose the development of a ground truth using pairwise human judgments.

To establish a ground truth dataset for the calculating design variety, we first need three components:

1. A ground set of design items over which sets are created
2. Sets of designs derived from the ground set for which variety scores are calculated
3. Tree annotations for each design item so we can calculate tree-based metrics

Variety scores are calculated on a set of designs. However, human raters are not good at giving absolute scores [146] due to differences between internal scales of subjects, a well-known problem for subjective pairwise scaling. For instance, given the set of designs shown in Fig. 2.14, it would be difficult for a human rater to say whether this set of six designs scores 6 out of 10 or 8 out of 10 for variety. Different raters may also use different internal scales.

In contrast, if we ask a rater to rate whether they find the variety of set shown in Fig. 2.14 Set A greater than the variety of those shown in Fig. 2.14 Set B, they may answer it relatively easily. Hence, we propose that ground truth for variety should be created using pairwise queries, where each query contains two sets and one set is voted by human raters to have higher variety compared to the other set. To elicit responses from experts, we give them two sets at a time and ask them for pairwise comparisons of the form: "Which set of designs has higher variety?"

2.5.2.2 Measuring Variety for Polygons

In this experiment, we compared the performance of HHID, SVS and NM metrics in measuring the variety of a set of polygons. We first create a base set of 27 polygons. Each polygon has three attributes — shape, color, and shading. Each attribute can take three unique values. Polygons can be rectangular, triangular or oval shaped. They can be red, blue or green colored. Shading varies between polygons as complete fill, shaded or empty.

The number of possible sets of polygons is very large (2^{27}), hence calculating the variety score of all possible sets is not feasible. Instead, we narrow down our search to focus on three set sizes: when the number of items in a set is 4, 6 and 8. If we ask human raters to compare sets with larger than eight items, the task becomes very difficult for them (due to the need to remember large number of items while deciding). For a given set size, we first randomly pick 100 sets for comparison. From these 100 sets, we calculate all possible pairwise comparisons (4950 comparisons). Next, we calculate SVS, NM, and HHID scores for each set in each pairwise comparison (*i.e.* scores for $4950 \times 2 = 9900$ sets). Without loss of generality, we assume that 'Color' is the functional principle, 'Shape' is the working principle and 'Shading' is the embodiment for SVS and NM metrics.

Result 1: Existing metrics cannot distinguish between sets. If we pick two random sets from all possible 2^{27} options, then it is less likely that the two sets will have the exact same variety score. Table 2.4 shows the percentage of comparisons where each metric

finds both the sets of equal variety. We note that SVS and NM metrics do not distinguish between a large percentage of comparisons (up to 37%), while HHID gives identical scores to a much smaller percentage of pairwise comparisons. This implies that existing metrics are not sensitive or discriminative to differences between sets.

Result 2: Existing metrics vote similarly to one another. Table 2.4 also shows the percentage agreement between different metrics. We see that SVS and NM vote similarly for 80-85% of set comparisons for various set sizes. This means that for a large proportion of comparisons, both metrics are indistinguishable as they give the same pairwise response. If SVS finds Set A has higher variety, then so does NM. In contrast, the agreement between HHID and other metrics is close to random. Due to the lack of benchmark dataset, it is difficult to comment on whether a lack of agreement between metrics is a good thing or not. We show later in the results, HHID aligns with the human raters more than SVS and NM for two datasets.

To establish a ground truth for comparing different metrics, we proceeded with the following steps. First, we selected pairwise comparisons where SVS and NM could distinguish between the two sets; that is, both the metrics did not calculate the same variety score to both sets. This is important since we want any collected human judgment to differentiate existing metrics, and we cannot do this if we select comparisons where the two metrics calculate the same value. Secondly, we select the sets where both metrics disagreed on their vote. This means if SVS voted Set A to be higher variety, then NM would vote Set B to be higher variety. Note that this is a small set of pairwise comparisons — as we noted from Table 2.4, both metrics vote similarly for more than 80% of

	Same Score			Agreement		
Method	SVS	NM	HHID	SVS-NM	HHID-SVS	HHID-NM
Size 4	27.3%	37.0%	15.8%	84.4%	54.2%	50.2%
Size 6	31.7%	21.4%	14.7%	81.0%	47.6%	50.0%
Size 8	28.5%	12.9%	10.9%	82.5%	49.4%	56.9%
Size 10	31.2%	14.5%	9.2%	84.4%	54.2%	50.2%

Table 2.4: a) Percentage of pairwise comparisons when design metrics give same score to both designs. Lower percentages are desirable as it indicates that a metric can distinguish between sets. We notice that SVS metric gives same score for approximately 30% *of the sets. b) The right side shows agreement between metrics for pairwise comparisons. We notice that SVS and NM vote similarly for more than* 80% *of the sets.*

the comparisons.

Finding human annotations for such sets allows us to find out which of the two metrics better aligns with human responses. Finally, we take the top 5 sets where SVS is most confident that one set has higher variety than another and the top 5 sets where NM is most confident that one set has higher variety than another set (*i.e.*, the difference between the scores are maximum). We combine these two to generate 10 queries which are then given to human raters.

To find the ground truth for polygons, we conducted an Amazon Turk study, in which we collected responses from crowd workers for pairwise queries. A sample query with two sets of eight polygons is shown in Fig. 2.13. Judging the variety of polygons does not require expertise in the area and Amazon Turk allows us to gain a large number of responses. We collected pairwise responses for three different set sizes. For each set size, we created ten pairwise queries. For each query, we collected ten responses from

Amazon Turk participants. We randomized the order of the queries and also the order of the options shown to different participants to reduce the possibility of any ordering bias. We subdivided the surveys into two parts to reduce fatigue. No worker was repeated across surveys. We repeated six queries to check the internal consistency of workers and filter out responses.

Result 3: Human raters largely agree on what it means to have a high variety set of polygons. The survey responses showed that on average people had consensus on one set being more diverse or higher variety than another set. Out of ten votes, the number of votes received by the set pairwise query receiving a majority vote for sets of size 4 was:[9, 8, 9, 7, 6, 9, 8, 6, 8, 7] respectively. This means that for the first query, 9 people out of 10 voted for the same set. For the second query where two sets of size 4 were shown, 8 people voted for the same set as being of higher variety. Similarly, for sets of size 6, [5, 5, 9, 9, 9, 8, 6, 8, 5, 8] votes were received by the majority set and [7, 5, 7, 7, 9, 9, 8, 6, 7, 6] votes were received by the majority set for sets of size 8.

A direct comparison between SVS, NM, and HHID metrics using the published weights would be unfair to SVS and NM, as HHID weight parameters can be optimized specifically for each domain. The published weights for SVS metric is [10, 6, 3, 1] and published weights for NM metric is [10, 5, 2, 1]. Hence, we give the same flexibility to SVS and NM metrics by allowing the weights of functional principle, working principle and embodiment to be optimized to maximize their performance. For a given metric (say SVS) and weight combination (say 4, 3, 3), we calculate the variety scores for both sets in a given pairwise comparison. Suppose we had total 10 humans who voted on a pairwise

comparison. If SVS metric finds that Set A has more variety than Set B, and 8 humans had also voted this way, we allocate all these votes to SVS metric. If the metric found Set B has higher variety than Set A, then this metric receives the 2 votes which humans gave to the other set. As we ask 30 different queries from people, to judge the metric, we aggregate votes for all 30 queries.

For our experiment, the maximum number of votes that any metric can receive is 220 — that is if it always votes with the majority opinion of human raters. Note that in an ideal world, if all humans voted for the same set for all 30 queries, the maximum number of votes that any metric can receive would be 300. Suppose a metric receives 200 votes in total, then we say that it has 90.9% alignment (100x200/220 = 90.9) with human ratings.

Result 4: HHID outperforms SVS and NM w.r.t. human agreement on polygon variety. Table 3.5 shows the comparison between SVS, NM, and HHID for alignment with human ratings. We find that SVS and HHID have similar best case performance. We varied the weights of each functional level between 1 to 10 in steps of 1, giving us 1000 possible performance scores corresponding to each weight combination [w1, w2, w3]. We find that HHID performs better than SVS in the median case. The median case is calculated over all thousand weight combinations.

From Table 3.5, we can conclude that HHID aligns with human perception of variety to the highest degree, irrespective of the choice of weights — that is, its performance is robust to weight choices. Even in the worst case, HHID aligns with 74.5% of human ratings. We find that the highest performance is obtained for many combinations of weights like 1, 2 and 10. SVS performs similarly, however, we generated these comparisons such

that SVS has high confidence in its choice between both the sets (by design). In contrast, if we select sets to compare at random, SVS calculates the same score for more than one-fourth of the queries. This drastically reduces the SVS performance in alignment with human responses — humans would have a clear preference between the variety of two sets, but SVS would be indifferent. Hence, the HHID metric outperforms both SVS and NM in alignment with human's judgment of variety.

Method	Median Case	Best Case	Worst Case	Sample optimal weights
HHID	81.8%	95.4%	74.5%	1, 2, 10
SVS	79.0%	95.4%	59.0%	2, 1, 1
NM	54.5%	86.3%	40.9%	10, 3, 1

Table 2.5: Alignment of different design variety metrics with human responses.

2.5.2.3 Measuring Variety for Milk-Frother Sketches

To measure the variety of milk-frothers, we gathered data from a previous experiment conducted by Starkey, Hunter, and Miller [243], which consisted of 934 ideas. Specifically, the data set consisted of ideas developed by 89 first-year students from an undergraduate engineering course and 52 senior students from a capstone engineering course including 95 males and 46 females. The ideas developed in this dataset were from a design task where participants were asked to generate ideas for a "novel and efficient milk-frother." This task was selected because the task addressed solving a product-based problem.

To create the dataset of sets of milk-frother sketches, we used the ground set of ten design sketches studied in Research Task 1 (Fig. 2.8). The benefit of using these ten sketches is the availability of tree annotations as well as information in the form of

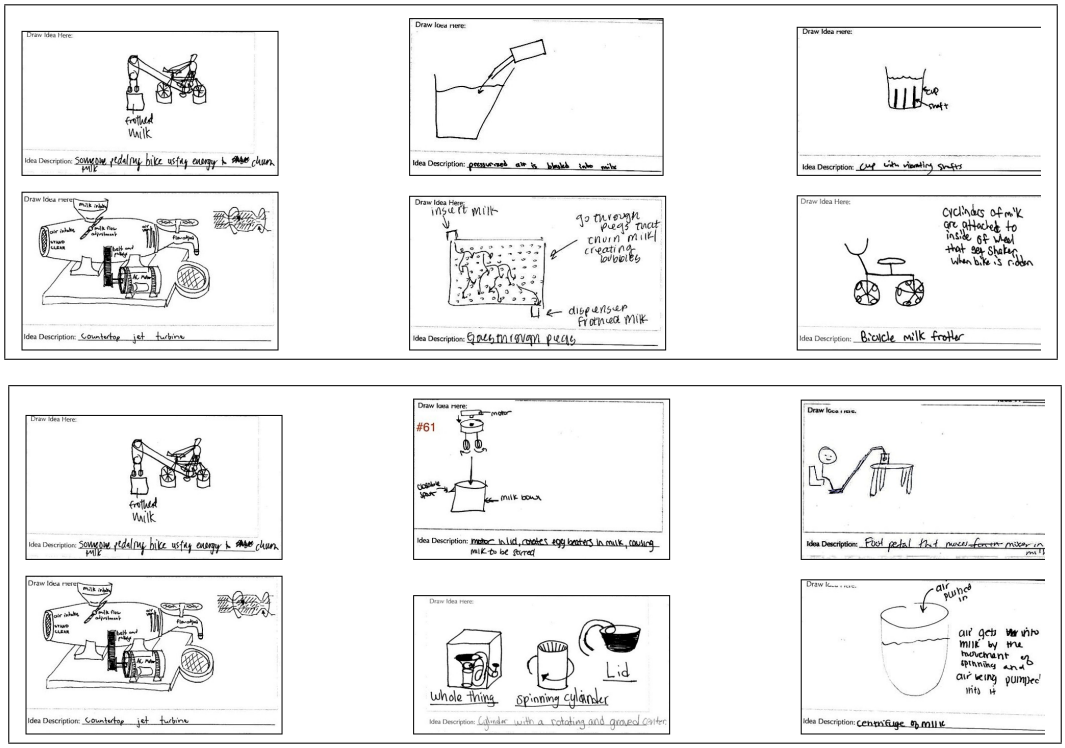

Figure 2.14: Top: Sample of Set A where all raters agreed it was more diverse than Set B. Bottom: Sample of Set B where all raters agreed it was less diverse than Set A.

subjective idea maps, which we use when discussing the final results below. The total number of possible sets for these ten sketches is 1024. We first calculate the variety scores for all these sets using SVS and NM metric.

Similar to the polygon case, to create a ground truth dataset of pairwise queries, we first want to find the queries which are most meaningful. However, in this case, we also have information about Euclidean embeddings for each sketch as discussed in Research Task 1 of this chapter. These embeddings are essentially 2-D maps with each design having x and y coordinates allocated to them. Similar designs occur closer to each other than dissimilar designs on this map. To find which sets to ask humans to rate, we use three metrics: SVS, NM and average pairwise distance of a set. The last metric is derived

using an embedding of designs derived in Research Task 1 of this chapter. The design embedding was picked randomly (as each participant in the study had a different design embedding and we needed only one design embedding to guide our experiment) and it provides the 2-D positions for each sketch and is only used to guide the selection of sets to be shown to human judges. The choice of the design embedding does not alter the key findings of this section as it is only used to guide the selection of queries which are asked from people. Using these ten sketches, our goal is to create pairwise queries with sets of six sketches each. We decided to create the ground truth with pairs of six images as the median number of sketches made by a participant in our dataset was six. The number of sets of size 6 is 210 unique combinations. We calculated the variety scores for all combinations and rank ordered them from the highest variety set to the lowest variety set using the pairwise average distance metric.

Out of these 210 sets, we obtained 21,945 pairs of sets (210 choose 2) and calculated the absolute rank difference between the two items for each comparison. A small rank difference implies that the two sets have similar variety score by a chosen metric, while a large rank difference implies that the chosen metric is confident that one of the set has a significantly higher variety than the other. After calculating the rank differences, we selected 20 comparisons based on two factors. First, we should select comparisons where each metric (pairwise distance, SVS, and NM) votes differently on which set has higher variety — *i.e.*, if all ratings agree on the comparison, then human expert ratings would not discriminate them. Second, we should select sets with a high-rank difference, but that also differ from sets we are using in other selected comparisons. That is, we want to ensure that a metric is confident in its vote, but that we also get good coverage over

different types of sets in the data by ignoring pairs that have already been selected.

Among these candidate sets, we select 20 pairwise queries that are given to four expert raters using a Qualtrics survey. We repeat two comparisons (10% repeated queries) in each survey to measure the internal consistency of each expert, giving them a total of 22 queries. Experts can choose whether Set A is higher variety compared to Set B or they can select the option of 'Can't decide'. From these expert ratings, we find that all four experts agreed on 9 out of 20 queries, while at least three experts agreed on 15 queries. Due to a majority agreement on these 15 queries, we select them as the ground truth dataset for comparing variety metrics. Next, we use this ground truth dataset to compare the SVS and NM metrics.

Result 5: SVS and NM are equivalent to random chance, w.r.t. matching expert assessments of milk-frother variety. We find that both SVS and NM align with only one-third (33.3%) of our human-provided ground truth dataset — that is five comparisons. We also change the weights for SVS and NM and report how close these metrics are to human experts. To explore the sensitivity of these results, we calculate the NM and SVS scores for every valid weight combination used by each metric. Using these weights, we find that SVS aligns with maximum 33.3% of the pairwise expert assessments of milk-frother variety irrespective of the weights used — that is, changing the tree weights used by SVS has zero effect on improving metrics agreement with human experts. NM aligns with 33.3% of the dataset for 95.6% of all the weight combinations. For the rest, it has no alignment with any expert ratings — that is, NM's scores are more sensitive to its internal weights, but not in a way that benefits its score accuracy with respect to human raters.

The alignment scores are close to random chance for three categories (Greater, Smaller and Equal) showing that SVS and NM are unable to capture human intuition of variety for the examples we tested.

Result 6: HHID robustly outperforms SVS and NM w.r.t. human comparisons, but still has a non-trivial error. In contrast to SVS and NM, HHID aligns with 9 out of 15 comparisons when weights are optimized for each level. We find that many weight configurations for HHID lead to highest performance (*e.g.* w=[1, 9, 5] leads to highest performance).

Hence, HHID aligns with human judgment of variety more than both SVS and NM metrics for two standard datasets. However, it still is not 100% accurate with respect to human benchmarks. However, we had assumed that the annotations provided for SVS, NM, and HHID for different hierarchical levels are accurate. If this is not the case, any variety metric will have a large error as it may not capture the true factors based on which humans decide their answers. Constructing the hierarchical trees is outside the scope of this work but it is important to understand that metrics may be limited by the specific choice of how one constructs a tree, which also needs to be verified.

We propose that by using our above method for constructing these ground truth variety comparisons, future work will be able to use these and other ground truth variety pairwise comparisons to judge the comparative quality of other metrics as well. This would provide a common scale over which metrics are compared.

2.5.2.4 Finding Sets of Designs with Highest Variety

One of the auxiliary outcomes of using an HHID derived index for variety measurement is that it provides a simple method to find the highest variety sets. Suppose you want to find a set of five polygons which have the highest variety from a given set of 27 polygons. Using existing NM and SVS metrics, the only way to do so is to enumerate all 80730 (27 choose 5) possible sets of five polygons, then calculate their NM and SVS scores and find the set with the highest score. This approach becomes infeasible when the ground set becomes large (for example 2.5 Billion sets for 200 designs) due to a large number of possible options (mathematically, this is because the problem is NP-Hard).

In contrast, we use Algorithm 2 to rank order all polygons or to select a subset. The resultant set is shown in Fig. 2.15. The set has a high variety score with respect to color, shape, and shading. The method selects one polygon at a time based on which polygon provides lowest marginal gain. As mentioned above, this is possible in polynomial time due to the supermodular behavior of HHID.

2.5.3 Discussion

Our experiments highlight several broader implications, both around how variety metrics are constructed and verified, as well as in how existing metrics are used across domains.

2.5.3.1 Assumptions and Limitations

Before adopting this methodology, one should be aware of various assumptions and limitations. This work makes the following assumptions:

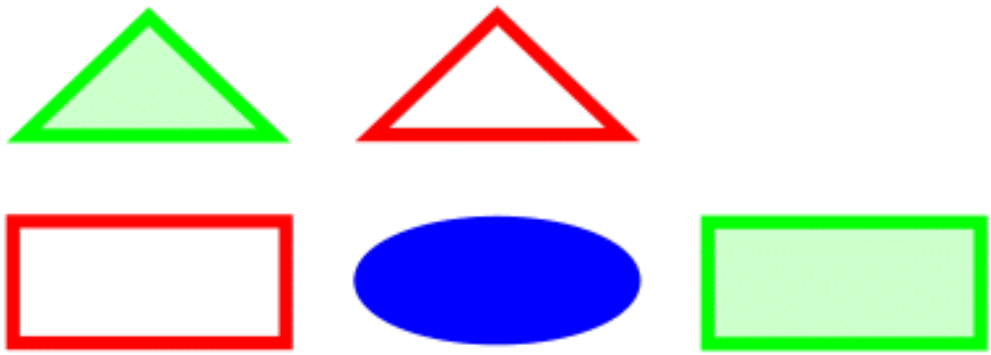

Figure 2.15: The set of five polygons with highest variety found using a greedy algorithm applied to the supermodular objective function capturing diversity.

1. We assume that the hierarchical principles (*e.g.* functional, working, embodiment) are correctly encoded and capture the attributes based on which the designs differ.

2. We assume that in deciding which set is more diverse (or has higher variety), people predominantly use the attributes which we encoded for each design. This means that suppose we encode energy, method of frothing and shape as the attributes, then people also use these three factors in their decision making. However, if people only consider color of the milk-frother in deciding which item has higher variety, our metric would not have enough information to measure the variety of sets.

3. In HHID, we square the number of items from each type of attribute. However, it is possible to use any other exponent, which is greater than 1 (the function retains supermodular properties for all values greater than one). In future work, one can optimize what power should be used within the metric for any given domain.

Next we discuss the key takeaways from our work and the main limitations which one should be aware of.

2.5.3.2 Selecting appropriate validation sets for variety measures is non-trivial

As we showed above, selecting exactly which sets of designs to show experts for ground truth labeling is non-trivial. First, the combinatorial nature of the problem (sets of designs) makes exhaustive labeling by experts impractical for anything above a handful of designs. But randomly sub-sampling this combinatorial set does not solve the problem: many metrics may trivially agree on a large portion of the space.

We proposed possible desiderata on what comparisons to show experts, as well as several potential methods to make this selection, such as maximal rank order disagreement, distances over embedded spaces computed via past techniques [18], and space coverage over different sets. Constructing comparisons in this fashion does lead to potential bias: as we saw in Result 4, by preferentially sampling sets where metrics were confident in their answers, we may, in fact, overestimate their performance with respect to their average performance in practice.

The trade-off here is one of time and cost. If one picks comparisons to maximize discriminative power among metrics, this will inevitably ignore portions of the space where they agree and inflate performance measures. In contrast, if one does not do this one may collect many expensive expert comparisons that, while covering the space well, do not provide much value in separating good metrics from bad ones.

One limitation of our proposed approaches is that we currently provide no theoretical guarantees regarding the number or scope of queries needed to achieve a certain assessment accuracy. The number of comparisons we collected above was driven by primarily practical concerns — how many expert comparisons could we realistically expect to collect in our available time budget? Future work could address how to perform this collection in an optimal fashion (*e.g.*, using Active Learning) and to bound the number of comparisons one would need to collect.

2.5.3.3 Good variety metrics need to be accurate and discriminative

As we showed in Results 1 and 2, good metrics need to not only be accurate but also highly discriminative or sensitive. We found that commonly used metrics can lack sensitivity across a broad range of comparisons. Even if such metrics are accurate, they have limited usefulness as measurement instruments — that is, they cannot detect small effect sizes in terms of differences in variety. We argue that, in addition to focusing on accuracy, future metric development should compute and account for the sensitivity of the measurement instrument for the given domain, and such quantities should be reported in subsequent research.

2.5.3.4 Metric performance can differ significantly across domains

Comparing Results 4 and 5, we see that a given metric applied to one domain/problem may have drastically different performance. In our case, SVS performed well with respect to human comparisons on the polygon case, but poorly on the milk-frother case. While it is perhaps obvious that a metric's accuracy depends on where it is applied, we note

that, in practice, past researchers have broadly used existing metrics (both SVS, NM, and others) with limited to no verification and calibration of the measurement instrument to that domain.

We believe that our results here should give other researchers pause before blindly applying an existing variety metric to a new problem without first conducting some of the pairwise verification we detail above. We are releasing both the datasets we collected in this work and the tools we used to construct human comparisons in the hope that future researchers will have an easier time constructing verification tests for new metrics or domains.[7] We believe that the proposed metric can be used in combination with other design metrics to provide insights from different perspectives of a set of designs. The usage of this metric and creation of new ground truth datasets should take into account the context that designers have deep knowledge in a field and can judge variety through different lenses and with an experience that may not always be possible from a quantitative metric.

2.5.3.5 HHID is a promising alternative metric that allows optimization of variety

We demonstrated via Results 4 and 6 that using HHID matched or exceed the performance of commonly used metrics. This was true in both the Polygon and Milk-Frother experiments. Calculating the HHID is computationally simpler to the benchmark tree-based constructions of SVS and NM.

More importantly, the supermodular form of HHID allows us to efficiently (*i.e.*, in polynomial time) approximate the highest variety sets of designs, given a corpus. For

[7] https://github.com/IDEALLab/design-variety

design corpora larger than approximately 50 designs, this leads to order-of-magnitude reductions in computational effort in finding optimal variety subsets of design, compared to existing metrics. The fact that HHID can be easily optimized to match human judgments for a domain makes it flexible to apply to different problems if one gathers pairwise comparison data as described above. Future work could cast the fitting of HHID as an active learning problem to reduce the number of expert comparisons needed to fit HHID to a given domain.

An important limitation of HHID and other tree-based metrics is that they are designed to measure variety for a set of designs with categorical attributes. HHID assumes attributes are in discrete space (*e.g.* categories of functional principle) and each category is independent of the other. For designs with attributes in continuous space, the metric does not work directly. In such cases, one may do clustering as a pre-processing step or explore other diversity metrics like Determinantal Point Processes which use positive semidefinite kernels to measure similarity between items.

2.5.4 Concluding Remarks of Research Task 2

In this task, we contributed: (1) a new design variety metric based on the Herfindahl index; (2) a practical procedure for comparing variety metrics via constructing ground truth datasets from pairwise comparisons by experts; and (3) empirically demonstrating the procedure and metric on two new two ground truth datasets using milk-frother design sketches and polygons. Overall, we provide a methodology of how validity of variety metrics should be assessed and then show how a new metric has higher validity than the alternatives. Using this dataset, we then compared the performance of two existing and

commonly used tree-based metrics and showed that our newly proposed metric aligns with human ratings more than existing metrics. As an ancillary benefit, we also show that by using a simple greedy algorithm our new metric can find sets of designs with the highest variety in polynomial time.

2.6 Key Contributions

The key contributions of this chapter are:

1. When design attributes are not available, we showed how design embeddings can be derived using simple to do triplet comparison tasks. Obtaining the embeddings allow for developing new metrics for novelty measurement of design items.

2. We provided a methodology to unpack subjective similarity decisions using design embeddings by looking at known attributes for each design and unpacking which attributes were more relevant in decision-making.

3. We established a procedure to verify design embeddings and novelty computation from them.

4. We proposed a new variety metric based on the Herfindahl–Hirschman Index and showed that it had better alignment with human judgments of variety compared to the alternatives ([235] and [193]).

5. We showed that by choosing a variety metric function which is monotone non-decreasing and supermodular, we can use a scalable greedy optimization algorithm with a constant factor guarantee of optimality to find sets of highest variety. The

greedy algorithm makes locally optimal choice at each step and guarantees that the final solution's variety will be atleast 0.63 of the highest variety solution. This allows us to find sets of ideas with high variety from a large collection in polynomial time.

6. We showed that SVS and NM metrics give the same variety score to a large percentage of sets, while HHID index has higher sensitivity in distinguishing between different sets of ideas. This shows that SVS and NM lack sensitivity.

2.7 Directions for Future Work

In the first research task, we computed idea maps from pairwise comparisons and showed that novelty metrics can be derived from these idea maps. Future work can explore many extensions of this direction of work. For instance, one can explore how to use active learning to extend this method larger datasets with fewer triplet queries. One can also try to code external human preferences into the optimization framework to find design embeddings with more information. Finally, while the methods discussed so far aim focused on single-view embeddings, the method can also be extended to multi-view embeddings, to obtain idea maps corresponding to each factor considered by the participant and calculate feature specific novelty. In the second research task, we highlighted existing problems with two commonly used variety metrics — SVS and NM. Next, we proposed a new entropy-based metric, which is more accurate and sensitive. It allows us to optimize variety as well as learn the weights for any new domain. Future work can focus on generalizing the HHID metric, learning it from a few data points as well as asking a few

queries to establish the ground truth. Below is a list of future directions of research:

1. New metrics: Compare different novelty metrics on idea maps and find alignment with human understanding of novelty. This will include non-metric methods of novelty computation too.

2. Improving scalability: Use active learning to reduce the number of comparisons asked from raters. Queries can also be reduced by using better preference elicitation methods like one to many triplet comparisons and new clustering interface.

3. Ratings: How many comparisons are required to reliably estimate novelty from idea maps?

4. Multiple views: Use multi-view triplet embeddings to find idea maps and novelty score corresponding to different viewpoints.

5. Experts vs novices: Unpack differences in how experts rate items compared to novices. Measuring differences between raters can help in training them too, by understanding what features someone is not paying attention to and providing appropriate intervention to increase inter-rater reliability.

6. Auxiliary information: Change the optimization method to accommodate additional information from rater about the design domain like the number of clusters or most novel item.

7. Combining human similarity judgments with attributes: Human insight can capture relationships that are not captured from the attributes. However, machines

can help relieve the human from having to exhaustively specify many constraints to learn all design attributes. Combining the two methods can allow to get the overcome the disadvantages of each method. One possible direction of doing so is using SNaCK [274], which combines the two methods to find low dimensional embedding for images. The method called "SNE and Crowd Kernel Embedding" combines expert triplet hints with machine assistance to efficiently generate concept embeddings.

8. Rater importance: In our research, we do not use explicit criteria to filter out raters with lower scores on rater reliability metrics (like self-consistency) but this information could be incorporated in future studies to give more importance to idea maps of raters who are more self-consistent.

9. Optimal queries for variety: How to perform the collection of comparisons to be given to experts in an optimal fashion (*e.g.*, using Active Learning) and to bound the number of comparisons one would need to collect.

10. Supervised learning of HHID: How to fit HHID metric to reduce the number of expert comparisons for a given domain?

11. Generalization of variety metrics: HHID metric is a special case of entropy and belongs to the family of Sharma-Mittal entropy functions. An area of future work will be to learn how one can fit the Sharma-Mittal entropy to a particular domain with minimum queries.

2.8 Conclusion of Chapter 2

At the beginning of this chapter, we asked the question: "How does one reliably measure the creativity of ideas?" To address it, we identified issues of validity, explainability and repeatability with two design metrics related to creativity — novelty and variety, and proposed computational metrics to address those issues. The goal was to help the design community measure novelty and variety of items in a principled way, backed by mathematical models as well as human intuition.

In the first research task, we addressed novelty measurement for ideas which are not directly represented in vector space. We showed how human subjective comparisons can be captured by triplet embedding methods, providing a doorway to novelty detection methods based on embeddings. In the next research task, we contributed a practical procedure for comparing variety metrics and proposed a new variety metric, which improved on the accuracy and sensitivity of existing metrics. Overall, our results shed light on some qualities that good design variety metrics should possess and the non-trivial challenges associated with collecting the data needed to measure those qualities. These results provide guidance on how and when various commonly used metrics may or may not be valid, as well as a concrete scientific process by which to gain further insight into when and where metrics apply.

We hope that the procedures we outline here can provide a catalyst for deeper discussion regarding how we measure and verify creativity metrics. We encourage researchers to build upon and contribute to the datasets we have started collecting and distributing for these problems. Our hope is that by better understanding how to measure the

variety and ultimately optimize variety, we will be able to reliably and scalably support designers in improving their creativity and competitiveness.

Overall, we unpacked how one can reliably measure the creativity of ideas, with the goal that a design contest organizer can use standard metrics to rate ideas submitted to a contest. Assuming an organizer has collected thousands of ideas, the goal of the next chapter is to allocate reviewers to each idea from a pool of available reviewers. However, at a more fundamental level, it adopts a quadratic-based diversity metric (similar to HHID discussed in this chapter) and proposes optimization methods to maximize diversity under different types of constraints. Specifically, in the next chapter, we study how diversity can be incorporated as an objective in a bi-partite graph matching problem and discuss algorithmic approaches to solve them. We show that one can form teams of reviewers by solving an Integer Programming problem and propose new objective functions and optimization methods to solve seemingly intractable problems.

Chapter 3: Diverse Team Formation: How does one form teams to evaluate design ideas?

Measurement and optimization of diversity metric is key to many domains. In this chapter, we develop fundamental methods to measure and optimize diversity metric while forming a team. Specifically, we investigate the problem of forming diverse teams *i.e.* given a set of workers and a set of tasks, allocate tasks to workers such that each task receives a diverse team of workers. This problem (often termed as a bipartite matching problem) generally assumes that each task needs a minimum number of workers and team size cannot exceed a maximum threshold. Similarly, each task needs a minimum number of workers and won't accept more workers than a maximum threshold. Our contributions in this chapter are in proposing computationally tractable algorithms to find teams which encourage diversity. We divide this chapter into three tasks:

- Research Task 1: Offline diverse matching: We propose an optimization-based approach to forming diverse teams when the entire pool of workers is available at the time of task allocation. For example, the assignment of reviewers to conference papers.

- Research Task 2: Offline diverse matching: We propose the first pseudo-polynomial algorithm for forming offline diverse teams with a theoretical guarantee of reaching

the optimal solution.

- Research Task 3: Online diverse matching: We propose an algorithm to form diverse teams when the workers arrive one at a time and the task allocation needs to be done immediately on their arrival.

3.1 Background and Motivation

Collaborative work often benefits from having teams or organizations with diverse backgrounds and experiences [246]. For example, studies have suggested that there is a positive relationship between diversity in a firm's knowledge base and its capability to innovate. Firms that are technologically diverse are more innovative and survive longer [135]. Firms or teams with employee diversity are often considered to be more competitive since such teams make the firm more open towards new ideas and more creative [207] — for example, by increasing a firm's knowledge base and interaction between different competencies. As the cultural, educational and ethnic backgrounds among employees become more diverse, so does the knowledge base of the firm.

If we have a pool of workers, and we know who is available to join different teams, then the problem reduces to the mathematical problem of static *bipartite matching*: that is assigning a set of resources (people, in this case) to a set of tasks/groups (teams, in this case). If different people were better suited for some teams or tasks over others (say, they had a certain skill that was highly valued for a given team's task), then this is called *weighted* bipartite matching, such that we assign people to teams such that the assignment maximizes the overall weight (or quality) of the matching. In practice, people

can often be assigned to multiple teams or collaborative projects at the same time, up to some upper and lower limits (say, a maximum of b number of teams or tasks per person), which is referred to as *weighted b-matching*. The widely-studied weighted b-matching problem occurs in a variety of situations including team-formation, scientific peer-review (assigning people to review papers) [59, 177], or any other cases where a finite set of resources (*e.g.*, people, computers, vehicles) needs to be matched to another finite set of resources (*e.g.*, teams, tasks, trips) [130, 163, 88]. In such a matching market, the central goal is generally to maximize economic efficiency subject. For example, a firm might wish to maximize the number of open positions filled along with ensuring that the workers it hires should hold high quality. In contrast to weighted b-matching, forming and maintaining diverse and high-quality teams over time can be challenging, in large part because people (whether in traditional firms or online collaborative groups) join and leave the firm sequentially, over time, rather than as one large cohort or pool. This problem is referred to as the online weighted b-matching problem.

In the first two research tasks of this chapter, we study the problem of maximizing diversity along with quality for the offline matching problem. That means we want to encourage matching a *diverse* subset of people to teams — *e.g.*, teams where people are not only well-matched to the task but also have complementary expertise or relevant but different viewpoints. A representative example that this work considered is matching academic papers to possible reviewers. A paper might have the highest relevance to three reviewers who come from the same lab group, perhaps because they all published heavily in a similar area. Existing weighted bipartite b-matching (WBM) algorithms [58] would likely assign those three reviewers to the same paper. Is this outcome desirable? On the

one hand, yes, because they have expertise related to the paper. On the other hand, those reviewers would stress similar points, given their common background. So the paper may only improve in a narrow (albeit important) direction. What if we wanted to diversify the reviewer backgrounds — to find reviewers well-suited to the paper *and* complementary to each other? Ideally, the reviews would remain high quality but would cover different, complementary aspects of the paper. We propose algorithms to solve such offline diverse matching problem in the first two research tasks of this chapter.

In the third research task, we handle a problem with an additional constraint — we do not know ahead of time exactly which future people will be available and need to decide in the moment whether to assign a newly arrived person to a team — *i.e.*, we must match people to teams *online* rather than waiting to collect a pool of people and then matching everyone in that pool to teams in an *offline* fashion). We refer to this as *online, diverse, weighted b-matching*. This setting is particularly important in practical implementations of computer-supported collaborative work, where teams of people are formed to solve problems together.

Overall, the goal of this chapter is to investigate how we can compute diverse matchings under various constraints, how we can bound the performance of these algorithms and demonstrate the practical applicability of diverse matching in simulated and real-world experiments. In achieving those goals, we develop theory and applications on how to measure and optimize the diversity of a set of items.

3.2 Literature Review

Bipartite matching, where agents on one side of a market are matched to agents or items on the other, is a classical problem in computer science and economics, with widespread application in healthcare, education, advertising, and general resource allocation. A practitioner's goal is typically to maximize a matching market's economic efficiency, possibly subject to some fairness requirements that promote equal access to resources. A natural balancing act exists between fairness and efficiency in matching markets, and has been the subject of much research. Matching people to form diverse teams leverages the intersection of two past areas of research: the role of team diversity in collaborative work and how diversity among groups of resources is measured and used to form/match teams. In this section, we review related work for both offline and online matching algorithms as well as the role of team diversity.

3.2.1 Diversity in Teams

Building effective teams is often defined as "helping a work group become more effective in accomplishing its tasks and satisfying the needs of group members" [72]. Prior research has explored what constitutes a successful team [67], how teams develop [164], and how different selection criteria and competencies might lead a team to excel [166]. For example, forming effective teams often involves finding the right combinations of pre-existing knowledge and skills — like problem solving, communication competencies, decision-making, goal-setting, performance and workload management capacity [128, 134] — as well as balancing the diversity or similarity of worker skills [229], workers' attitudes,

personalities [33] and emotional intelligence [140].

Over the past few decades, a great deal of research has been conducted to examine the complex relationship between team diversity and team outcomes. The concept of "diversity" has a variety of meanings, including separation in attitudes or viewpoints; variety of positions, categories or backgrounds; and disparity in values on some resource or asset [118]. Before discussing the effects of diversity, one must understand its broad categorization into task-related diversity and bio-demographic diversity. Bio-demographic diversity represents innate member characteristics that are immediately observable and categorized (e.g., age, gender, and race/ethnicity) whereas task-related diversity is acquired individual attributes (e.g., functional expertise, education, and organizational tenure) that have been postulated to be more germane to accomplishing tasks than bio-demographic diversity.

Researchers have shown that different types of worker diversity have a direct impact on the success rate of tasks [222]. For example, firms with a higher number of employees with a higher education and diversity in the types of educations have a higher likelihood of innovating [200] and increasing revenue for firms [135]. Task-related diversity has been reported to have a positive impact on team performance although bio-demographic diversity is shown not to be significantly related to team performance [131].

There are two core theoretical perspectives for effects of diversity of interest: the similarity-attraction paradigm and information processing [186]. Similarity-attraction theory predicts that similar people will be attracted to one another and will like each other better than less similar people. The information-processing approach, however, argues that the benefit of access to people with diverse backgrounds, information, social

networks, and skills will out weigh the potential coordination challenges that can arise from diversity.

Non-diverse teams often suffer from the problem of overemphasis on consensus-seeking behavior, which can result in suboptimal decision making. Perhaps the most well-known example of this is "groupthink," which can arise when groups place too much importance on attaining consensus and fail to debate important alternatives for fear of damaging group cohesion. Team diversity can often circumvent such myopia by bringing in differing perspectives and promoting healthy debates and dissents [275]. Recently, [207] discuss how culturally diverse teams have the potential for enhanced creativity relative to culturally homogeneous teams.

For demographic diversity, initial negative performance effects appear to diminish over time (*e.g.*, [119]). For cognitive diversity, positive effects are more likely to ensue when tasks are complex and non-routine [210]. Using the theoretical arguments of the cognitive diversity hypothesis, several researchers have argued that team diversity has a positive impact on performance because of unique cognitive attributes that members bring to the team [69]. On the other hand, many researchers argue that homogeneous teams work well together because of their shared characteristics, thereby increasing team cohesion and performance [43]. Increased task diversity affects human motivation to complete tasks in both physical [113] and virtual workplaces [145]. For example, [22] show that having diverse group of micro-tasks contributes to improving outcome quality for their participants.

3.2.2 Measuring Diversity and Matching Teams

While researchers have found many benefits to encouraging different types of diversity (cognitive, task-based, *etc.*) when forming teams, one open question lies in how to actually rigorously and scalably *form* teams (or, equivalently, match people to teams) to encourage that diversity. To do this, we first need to understand two areas of related research: 1) how to measure different types of diversity and 2) how to then use those measures to match people into diverse teams.

Past researchers have generally measured diversity by defining some notion of *coverage* — that is, a diverse set is one that covers the space of available variation. Farhangmehr *et al.* [92] used a Shannon entropy-based metric to assess the quality of solutions obtained from multi-objective optimization algorithms. The same authors [93] also presented a new Entropy based Multi-Objective Genetic Algorithm which leads to better coverage of points on trade-off front. Mathematically, researchers have often measured coverage via the use of *submodular functions*, which encode the notion of diminishing returns [172, 173]; that is, as one adds items to a set that are similar to previous items, one gains less utility if the existing items in the set already "cover" the characteristics added by that new item. For example, many previous diversity metrics used in the informational retrieval or search communities — including Maximum Marginal Relevance (MMR) [49], absorbing random walks [289], subtopic retrieval [283] and Determinantal point processes [162] — are actually instances of submodular functions. These functions can model notions of coverage, representation, and diversity [14] and they achieve the best results to date on common automatic document summarization benchmarks — *e.g.*,

at the Document Understanding Conference [172, 173]. These functions are widely used in extractive document summarization [172] to get a diverse high quality summary of documents. We used a similar reasoning when defining our objectives for diverse matching.

However, the use of these functions is not limited to just documents; they can model coverage over any objects, so long as we have some way to mathematically describe their characteristics. For instance, [198] used submodular functions to optimize the diversity of which people worked on a task to minimize redundant information from people who give correlated answers. They found that managing and exploiting diversity within their model improved task accuracy (in their case, a crowdsourced prediction problem). Diverse matching can be broadly categorized into offline and online diverse matching. In offline matching, we assume that all workers are presented in a pool when task allocation is done, while in online matching the workers come one at a time.

3.2.3 Offline Matching

In practice, the weighted bipartite b-matching (WBM) problem — find the feasible matching with maximum weight — has arisen naturally as a problem in many fields, such as protein structure alignment [158]; computer vision [35]; estimating text similarity [202]; VLSI design [132]; and matching reviewers to papers in peer-review systems [59, 177, 254]. Driven by practical application, such previous work aimed to maximize economic efficiency.

To address the need to maximize diversity along with efficiency, we first incorporated diversity objectives into the WBM problem. In the past, only a few researchers

had studied diverse matching problem. For example, Liu *et al.* [177], performed a node-specific diversity-inspired pre-process before solving a related matching problem. In contrast, we considered the "global" diversity of the full matching, a function of the diversity of *sets* of vertices. In other domains, past research had addressed diversity in ranking problems (*e.g.*, diverse recommendations [4, 233, 29]), but not for matching. There are many application-dependent choices for what such a space entails including vector spaces such as text vectors [213] or metrics over graphs [284], among others.

Other researchers have also approached similar problems, with diversity either as an objective or as a constraint. Chen *et al.* [60] proposed a Conflict-Aware WBM (CA-WBM). They considered conflict constraints between vertices on the same side of a bipartite graph. In CA-WBM, if two vertices were in conflict, they may not both be matched to a vertex on the other side of the graph. CA-WBM enforced a kind of binary diversity by manually defining conflicts between specific nodes. In [105], the authors match migrants to localities in a way that maximized the expected number of migrants who find employment. The authors solved maximization of an approximately submodular function subject to matroid constraints. [36] studied the trade-off between diversity and social welfare for the Singapore housing allocation problem. They modeled the problem as an extension of the classic assignment problem, with additional diversity constraints. [168] solved the assignment problem when preferences from one side over the other side are given and both sides have capacity constraints. They used order weighted averages to propose a polynomial time algorithm which led to high quality and more fair assignments. [7] showed that a simple iterative proportional allocation algorithm can be tuned to produce maximum matching with high entropy. Finally, [27] addressed a complementary problem in

minimum color-degree matching, and give complexity results.

In contrast to above methods, we proposed a Diverse Weighted Bipartite Matching (named D-WBM) [10], which treated diversity as an objective, not a constraint, allowing us to flexibly control the degree to which a matching algorithm encourages or discourages diverse solutions to the standard WBM problem. This is useful when one wanted conflicts or diversity to vary in degree, or trade off diversity with other measures of match quality. The method solves a Mixed Integer Quadratic problem to provide diverse matching solutions. In our second task, we extended the D-WBM optimization based method [10] to address two key issues — how to improve scalability and how to give provably optimal solution. We showed that by creating an auxiliary graph and finding negative cycles on that graph, we can guarantee optimal solution for a variant of the diverse matching problem.

3.2.4 Online Matching

In online task assignment problems: 1) a firm has a fixed set of tasks/teams and a budget that specifies how many times the firm would like each task completed or how many teams it needs; 2) new people arrive at the firm one at a time (in the case of regular hiring) and potentially the same person could arrive multiple times (*e.g.*, in the case of freelancing or gig/shift work); and 3) people must be assigned to a team immediately upon arrival (or rejected and not assigned to any team). The goal is to allocate people to teams in a way that maximizes the value of collaborative work all teams produce (*i.e.*, solely maximizing utility).

Compared to past related work, our online diverse matching work provides a prac-

tical, simple-to-implement, and high-performing method to perform diverse, online b-matching that can enable diverse team formation when unknown people arrive sequentially over time. Online matching and its generalization to set packing have been studied through the lens of theoretical computer science for nearly three decades [144]. These algorithms have been applied to a multitude of tasks like online video summarization [188]. The algorithms we present in this work draw motivation most heavily from recent work in online stochastic optimization with nonlinear objectives [79, 6], and from [279] in particular.

The offline case of [11] provided an algorithm for *diverse* b-matching applied to reviewer-paper matching of conference papers. This work addresses that gap by contributing a means to form teams that are both diverse and formed in an online fashion. This work addresses that gap by contributing a means to form teams that are both diverse and formed in an online fashion.

3.3 Research Gaps and Research Objectives

In our literature review, we found four main research gaps which are addressed in this dissertation. First, we did not find a matching method which treated diversity as an objective. Second, there is a lack of matching algorithms with guarantees of reaching optimal solution for diverse allocation. Third, no metric is defined in the literature to quantify diversity utility trade-off. Finally, the existing algorithms in the literature on constrained online submodular maximization cannot be applied to form diverse teams in practical situations like workers arriving sequentially on MTurk.

To address above issues, we first propose two algorithms to solve offline weighted diverse b-matching using a supermodular function, which can be optimized using a quadratic integer program method or a greedy algorithm proposed by us. We use both capacity and cover constraints in this problem and demonstrate the efficacy of our results for paper-reviewer assignment and movie recommendations. We define 'Price of Diversity' metric to quantify the trade-off between diversity and utility. In the second research task, we propose another algorithm, which provides a pseudo-polynomial time algorithm for the diverse matching problem which guarantees convergence to the optimal solution. Finally, in the third part we propose an online algorithm where sequentially arriving workers are allocated to teams based on the marginal gain they offer to the teams, as measured by a submodular diversity function.

3.4 Research Task 1: Optimization-based Offline Diverse Matching

In this research task, we study the goal of balancing *diversity* and efficiency — in a generalization of bipartite matching where agents on one side of the market can be matched to *sets* of agents on the other. Adapting a classical definition of the diversity of a set, we propose a quadratic programming-based approach to solving a supermodular minimization problem that balances diversity and total weight of the solution. We also provide a scalable greedy algorithm with theoretical performance bounds. We then define the *price of diversity*, a measure of the efficiency loss due to enforcing diversity, and give a worst-case theoretical bound. Finally, we demonstrate the efficacy of our methods on three real-world datasets, and show that the price of diversity is not bad in practice.

3.4.1 Weighted Bipartite Matching

N	$\triangleq$	Number of nodes on the right side
M	$\triangleq$	Number of nodes on the left side
L^-, L^+	$\triangleq$	Minimum and maximum edges that can be matched to each left side node
R^-, R^+	$\triangleq$	Minimum and maximum edges that can be matched to each right side node
X	$\triangleq$	Column vector of binary variables of size MN. x_{ij} = 1 if the edge connecting left node i to right node j is present in the matching
A	$\triangleq$	$(M+N) \times MN$ sized linking matrix indicating which nodes are allowed to be connected to what nodes in the initial bipartite graph
U	$\triangleq$	Set of all left side nodes
V	$\triangleq$	Set of all right side nodes
E	$\triangleq$	Set of all edges
W	$\triangleq$	Weights of the edges with w_{ij} denoting the weight of the edge connecting left node i to right node j
K	$\triangleq$	Number of clusters into which left side nodes are partitioned
B	$\triangleq$	$MN \times MN$ sized block-diagonal matrix capturing the quadratic objective

Table 3.1: Table of Notation for Research Task 1.

Weighted bipartite b-matching is a combinatorial optimization problem formulated as follows. Given a weighted bipartite graph $G = (U, V, E)$ with weights $W : E \rightarrow R^+$, where U, V and E represent left vertices, right vertices and edges, the weighted bipartite

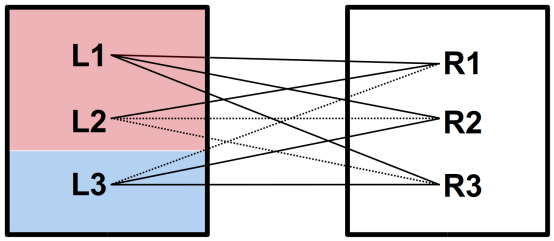

Figure 3.1: Bipartite b-matching problem where the left side nodes are divided into two clusters.

b-matching problem is to find a subgraph $T \subset G$ such that each vertex i in T has at most b edges (*i.e.*, a degree constraint). WBM maximizes or minimizes the objective depending on the application.

We use similar notation to Chen *et al.* [60] to define a weighed bipartite b-matching problem, with two notable differences. First, we define it as a minimization problem, and second, we define a harder problem which has both node-specific upper- and lower-cardinality constraints. The constrained weighed bipartite b-matching (WBM) problem can be expressed as follows.

$$\begin{aligned} \min_{X} \quad & f_1 = WX \\ \text{s.t.} \quad & L^- \leq AX_i \leq L^+ && \forall i \in \{1, \ldots, M\} \\ & R^- \leq AX_i \leq R^+ && \forall i \in \{M+1, \ldots, M+N\} \\ & x_{ij} \in \{0, 1\} && \forall i, j, 1 \leq i \leq M, 1 \leq j \leq N \end{aligned} \tag{3.1}$$

We have N items on the right side with R^- and R^+ integral lower and upper cardinality constraints, respectively, and M items on the left side with L^- and L^+ as integral cardinality bounds. Here, X is a column vector of binary variables of size MN, with $x_{ij} = 1$ if left item i is matched to right item j, and $x_{ij} = 0$ otherwise. W is a matrix of weights w_{ij} representing the local quality of matching items i and j.

Items on the same side of bipartite graph cannot be matched; thus, we use A as a linking matrix, such that any row i indicates which nodes are allowed to be connected to item i. We index edges (i, j) uniquely using a function $\ell : E \rightarrow \{1, \ldots, MN\}$. Then, A is an $(M + N) \times MN$ matrix, where $a_{i\ell((i,j))} = 1$ if edge (i, j) exists; otherwise, $a_{i\ell((i,j))} = 0$. The degree constraints for left nodes are given by $L^- \leq AX_i \leq L^+$, where AX_i denotes the i^{th} element in (vector) AX. L^+ is the upper bound on cardinality of i^{th} node and L^- is the lower bound.

The above formulation shows a constrained matching problem, where nodes on both sides have capacity constraints. This discrete linear optimization problem is NP-hard [60]. Its optimal solution will minimize the weights, emphasizing on efficiency and neglecting diversity.

3.4.2 Diversity in Offline Matching

Diversity in matching can be defined as the need to match a node with other nodes from different groups. To add diversity, we consider a scenario where left-side nodes are divided into K groups. Let us say that we want a matching which matches each node on the right side to nodes from different clusters. The diversity is calculated using supermodular functions. These functions have been widely used in extractive document summarization [172] to get a diverse high quality summary of documents. We use a quadratic function which can incorporate diversity by balancing the number of nodes (e.g., items or agents) selected from different clusters.

Let $E_l = \{(1, l), \ldots, (M, l)\}$ be the set of all M edges from a node $l \in \{1, \ldots, N\}$ on the right side of the graph. Let the subset $S_l \subseteq E_l = \{(1, l), ..., (m, l)\}$ be the matched

m edges for node l. Let $(P_i)^l$, $i \in \{1, \ldots, K\}$ is a partition of the ground set E_l into K separate clusters (*i.e.*, $\cup_i P_i^l = E_l$, and $\cap_i P_i^l = \varnothing$). That is, a left item can only belong to one cluster. The weight of an edge from left node n to right node l is $w_{n,l}$. We define the quality of match for node l on the right side as:

$$f(l) = \sum_{k=1}^{K} \left\{ \sum_{j \in S_l \cap P_k^l} w_{j,l} \right\}^2 \tag{3.2}$$

This quadratic function gives lower cost to solutions with even coverage over all clusters. As an example, Figure 3.1 shows three nodes on either side, each requiring two edges. If all edge weights w are one, the node-specific utility of a matching $\{(L1, R1), (L2, R1)\}$ is 4, while the utility of alternate matching $\{(L1, R1), (L3, R1)\}$ is 2. Hence, by minimizing the function in Equation 3.2, diversity is encouraged. By transforming the matching problem to quadratic minimization, the resultant objective function simultaneously optimizes quality and diversity. Keen observers may notice that the definition of diversity in Eq. 3.2 is similar in form to the definition of variety (HHID in Eq. 2.6) in Chapter 2, which was also a supermodular function. In the next section, we provide a formal framework to generalize this function to constrained b-matching problems.

To the best of our knowledge, no known general measure exists to measure the performance of diverse b-matching methods. Even verification of diverse matching is difficult, due to different definitions of diversity in the literature. One way of comparing our diversity results is to look at the Shannon entropy of a match for each item, with and without our method. Shannon entropy has been used to incorporate diversity in recommendations [216, 80] and also widely used in the ecological literature as a diversity index.

It quantifies the uncertainty in predicting the cluster label of an individual that is taken at random from the dataset. Here entropy of a node is given by: $-\sum_{i=1}^{K}(p_k \log p_k)$, where p_k is the proportion of selected edges in cluster K.

Entropy for an item is maximized if it is matched to other items with even coverage of different clusters; it is zero when all such items are from the same cluster. Hence, the impact of diverse matching can be measured as improvement in average entropy. We define the *entropy gain* (EG) as:

$$EG = \frac{\text{Average entropy using a diverse matching rule}}{\text{Average entropy using WBM}} \tag{3.3}$$

EG is large if the average Shannon entropy of diverse matching is large compared to the average entropy of non-diverse WBM matching. We also propose a new metric named *price of diversity* (PoD) to measure the efficiency lost due to diversity. We define the *price of diversity* (PoD) as:

$$PoD = \frac{\text{Utility using WBM}}{\text{Utility using a diverse matching rule}}. \tag{3.4}$$

A large value of PoD implies that the diverse matching has similar efficiency as the non-diverse WBM matching. Later in the work, we will show in simulation and on real data that the entropy gain under our proposed diverse matching method is high, at very little cost to overall efficiency.

3.4.3 Exact and Approximate Algorithms

In our bipartite matching formulation with utility minimization, the degree constraints L^- and R^- can be interpreted as setting the demand. The short side of market determines the number of edges in the matching, which is $\min\{ML^-, NR^-\}$. If the right side is short,

the maximum capacity on the left should be more than the demand: $NR^- \leq ML^+$ for any matching to be feasible. For the purpose of this work, we always assume that right side is the short side of market and the left side is clustered into groups. The cardinality constraints make the problem more difficult than what has usually been solved for matching as nodes cannot be matched independent of each other.

3.4.3.1 Diverse Weighted Bipartite b-matching

To generalize the quadratic function (cf. Equation 3.2) to an optimization framework for all nodes, we define a $MN \times MN$ block-diagonal matrix $B = \text{diag}(B_1, \ldots, B_M)$ such that:

$$B_l = \begin{cases} w_i \cdot w_j \text{ if edges } i, j \in P_k^l \text{ (same cluster)} \\ 0 \text{ otherwise} \end{cases}$$

B_l is the block diagonal matrix for every right node, with K blocks on the diagonal corresponding to each cluster. Matrix B is a diagonal matrix for all B_l matrices combined. Later, we show a visualization of the symmetric B_l matrix in Fig. 3.3 for five clusters for reviewer matching application. Hence, the optimization problem for Diverse WBM (D-WBM) can be written as:

$$\underset{X}{\text{minimize}} \quad f_2(X) = X^T B X \tag{3.5}$$

To show that this formulation is equivalent to Eq. 3.2, let us again consider $R1$ in Fig. 3.1 with two clusters, three edges and unit weights. Using Eq. 3.5 for the node-specific utility of a matching $\{(L1, R1), (L2, R1)\}$ is $[1,1,0]'[1,1,0;1,1,0;0,0,1][1,1,0] = 4$ and the utility of alternate matching $\{(L1, R1), (L3, R1)\}$ is $[1,0,1]'[1,1,0;1,1,0;0,0,1][1,0,1] =$

2. This is same as obtained by Eq. 3.2 before. The constraints and variables are the same as in WBM (cf. Equation 3.1). Our new model has a quadratic objective with linear constraints and integrality requirement for variables. We solve the quadratic objective optimization problem using two different approaches, first using Gurobi's Mixed Integer Quadratic Programming (MIQP) Solver [112], and second by using a novel greedy algorithm that builds up a set by minimizing marginal gain. We also solve the non-diverse WBM model using Gurobi mixed integer solver to compare it against diverse matching solutions. Next, we propose this greedy algorithm and give bounds on its performance.

3.4.3.2 Greedy Diverse WBM

The objective of the D-WBM formulation is supermodular, that is, adding new elements leads to (strictly) increasing differences. Hence a method which greedily adds edges by minimizing the marginal gain can attain reasonable performance bounds [194]. Secondly, solving the MIQP exactly requires storing the block diagonal matrix; for large problems, MIQP may run out of memory as the number of non-zero terms in the matrix scales by M^2N.

We propose Algorithm 2, which incrementally *satisfies* the lower degree constraints for all nodes. It does this by increasing the lower bound of all nodes unit step at a time and selecting edges by minimizing marginal gain in the objective f_2 (Eq. 3.5). In selecting edges, it gives preference to the set of nodes with unsatisfied lower bound. This ensures that the greedy selection always finds a feasible matching with good empirical performance.

For a right constrained matching, the matching for every right node is indepen-

Algorithm 2: GD-WBM Greedy Diverse Matching.

Input: A set of $N + M$ nodes, bounds L^-, L^+, R^-, R^+ and edge weights matrix B

Output: A feasible matching

$C \leftarrow \emptyset$

for $i \leftarrow 1$ ***to*** $\max\{L^-, R^-\}$ **do**

  $L_i^- = \min\{i, L^-\}; R_i^- = \min\{i, R^-\}$

  for $j \leftarrow 1$ ***to*** $N + M$ **do**

    if *j's lower bound is not satisfied* **then**

      Select the feasible edge e with lowest marginal gain

      $f(C \cup \{e\}) - f(C)$ whose opposite node's lower bound is not satisfied

      If no such node exists, pick the feasible edge with lowest marginal gain.

return C

dent of others as they do not have overlapping constraints. Hence GD-WBM achieves a $(1 - 1/e)$-approximation of the optimum due to its supermodular objective function. In practice, Section 3.4.4 will show that GD-WBM performs much better than the lower bound.

3.4.3.3 Price of Diversity Bound

In this section, we propose lower bounds on the price of diversity (PoD) — the utility loss due to diverse matching — in right-constrained market. Theorem 1 gives such a bound.

Theorem 1. *The worst-case Price of Diversity (PoD) for a right-constrained diverse matching is:*

$$\frac{1}{N}\sum_{l=1}^{N}\frac{z_l}{1+\sqrt{R^- - 1}\sqrt{{z_l}^2-1}},\ \textit{where}\ z_l = \frac{\sum w_{jl}}{\min w_{jl}} \tag{3.6}$$

Proof sketch. We wish to find a problem instance where the best diverse matching has high weight under the utilitarian objective. Such PoD for a matching will be minimized when diversity leads to all low-weight weight edges being replaced by higher weight edges. Such a situation can occur when WBM provides a match for a right node with all m edges going to left nodes in the same cluster 1. Let $\{w_1, \ldots, w_m\}$ be such edge weights. In this instance, let D-WBM select edges going to m unique clusters. Here, D-WBM will select the single edge with least weight from each cluster, including cluster 1. Call those edges $\{a_1, \ldots, a_m\}$ be the selected edge weights by D-WBM. The edge weights will satisfy the following constraints:

$$\sum_{i=1}^{m} w_i \leq \sum_{i=1}^{m} a_i \quad \text{and} \quad (\sum_{i=1}^{m} w_i)^2 \geq \sum_{i=1}^{m} {a_i}^2 \tag{3.7}$$

Both WBM and D-WBM select edge $a_1 = \min_{i \in [m]} w_i$ from cluster 1. To minimize PoD, the denominator — f_1 of the diverse matching — $\sum_{i=1}^{m} a_i$ should be maximized. Using the Lagrangian method, this constrained maximization problem is solved, with optima occurring at the surface of a hypersphere and $a_2 = a_3 \ldots = a_m$. □

If the minimum weight for every right node is 0, by taking limits on Eq. 3.6, the PoD becomes $\frac{1}{\sqrt{R^- - 1}}$. In the succeeding section, we will show that Theorem 1 is quite conservative.

3.4.4 Results and Discussion

We begin by comparing the two methods' PoD to its theoretical bound on a synthetic dataset. Next, we solve the b-matching problem on three datasets, one for movie recommendation and two for papers–reviewers matching. We analyze the effect of problem size by increasing the number of nodes and the cardinality requirements. Finally, we discuss the trade-off between diversity and utility.

3.4.4.1 Artificial Dataset

In this section, we simulate matching 10 nodes with another 10 nodes; by Theorem 1, the worst-case PoD is 0.5. Weights are selected randomly from a uniform distribution between 0 to 1 and the cluster labels of the left-side nodes are selected randomly. For right-constrained matching, we use $R^- = 5$, implying that each right side node will be matched to at least five items. The number of clusters are varied from 2 to 10, and 100 trials are done to compare D-WBM with WBM.

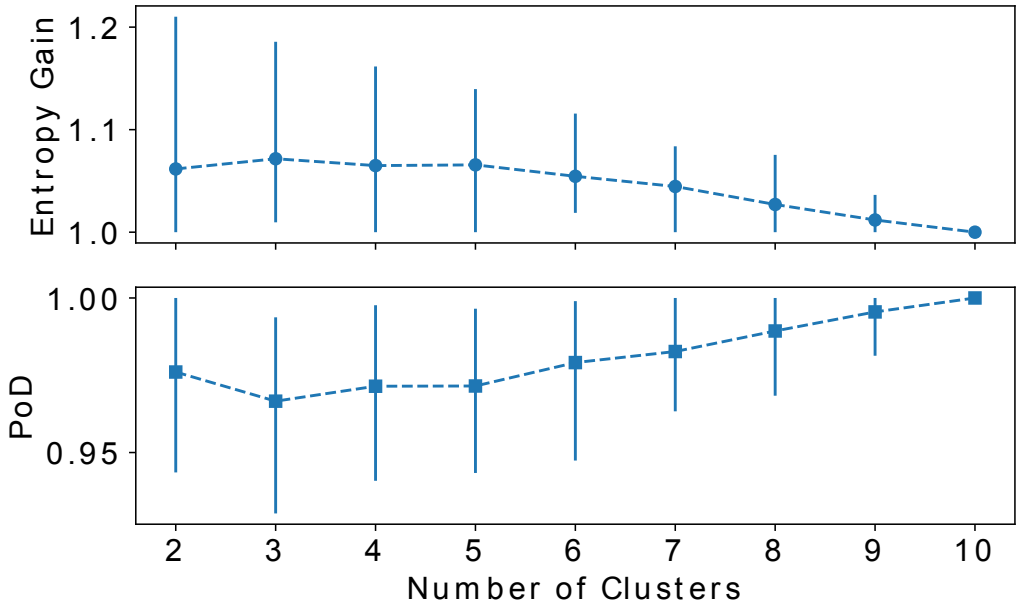

Figure 3.2: PoD and EG for a simulated dataset showing the average PoD with 5^{th} *and* 95^{th} *percentile values. D-WBM unilaterally outperforms the worst-case PoD of* 0.5.

Figure 3.2 shows that in practice, PoD is never below 0.9 despite random clusters and weights. Also, EG generally decreases when PoD increases. The greedy approximation GD-WBM finds the same matches as D-WBM for all cases.

3.4.4.2 Application to MovieLens Dataset

This example considers matching movies to users, while ensuring that the movies contain diverse genres. We use a subset of the MovieLens 1M dataset [117], which includes one million ratings by 6,040 users for 3,900 items. This dataset contains both users' movie ratings between 1 and 5 and genre categories for each movie (*e.g.*, comedy, romance, and action).

We first train a standard collaborative recommender system [40] to obtain ratings

for all movies by every user. We cluster the movies into 5 clusters using their vector of 18 genres using spectral clustering,[1] so that each movie gets a unique cluster label. We solve the right constrained matching problem for 500 movies and make recommendations for 500 users with at least 10 recommendations for every user. Table 3.2 shows that with average EG of 1.45, D-WBM finds a more diverse matching for all users and on average a user loses only 1% utility for this gain. To save computational time, in all our simulations we terminate D-WBM after 1 hour and take the best solution, while WBM converges to true optima. Hence the results are conservative estimates.

To further understand the matching result, we compare the movie recommendations by D-WBM and WBM for User ID 423. WBM matches her to movies that are all either Comedies or Dramas, with an average predicted rating of 4.07. In contrast, D-WBM matches the user with movies from five different clusters, with an average movie rating of 4.05, showing negligible loss in efficiency. Table 3.3 compares the recommended genres. While we chose to promote diversity in genre, the MovieLens dataset also provides information about the user's gender, age group, and occupation; D-WBM can encourage other types of diversity in movie recommendation.

3.4.4.3 Application to Reviewer Assignment

In this section, we present an application of diverse matching to automatically determine the most appropriate reviewers for a manuscript by also ensuring that reviewers are different from each other.

[1] The exact choice of recommender system and clustering algorithm is not central to this research; it just helps set up the graph.

Dataset	D-WBM		GD-WBM	
	PoD	EG	PoD	EG
Movie-Lens	0.99	1.45	0.99	1.45
UIUC Reviewer	0.92	1.63	0.83	1.60
Sugiyama	0.94	4.28	0.93	4.28

Table 3.2: Performance of algorithms on Price of Diversity and Entropy Gain metrics for three real world datasets.

D-WBM		WBM	
Cluster	Genres	Cluster	Genres
2	Drama, Thriller	2	Drama,Thriller
1	Adventure,Sci-Fi	2	Crime,Drama,Thriller
3	Documentary	0	Comedy
3	Documentary	0	Comedy,Drama
0	Comedy,Romance	0	Comedy,Romance
4	Drama,Horror	2	Drama
4	Horror,Sci-Fi,Thriller	2	Drama
2	Drama	2	Drama
0	Comedy,Mystery	0	Comedy,Mystery
1	Action,Thriller	2	Action,Crime,Drama

Table 3.3: Genres and cluster labels of ten movies recommended to a sample user by WBM and D-WBM. The movies allocated by D-WBM provide a broader genre coverage compared to WBM.

UIUC Multi-Aspect Review Assignment Dataset: We use the multi-aspect review assignment evaluation dataset [143] which is a benchmark dataset from UIUC. It contains 73 papers accepted by SIGIR 2007, and 189 prospective reviewers who had published in the main information retrieval conferences. The dataset provides 25 major topics and for each paper in the set, an expert provided 25-dimensional label on that paper based on a set of defined topics. Similarly for the 189 reviewers, a 25-dimensional expertise representation is provided.

For the reviewer-paper bipartite graph, edge weights between each test paper and reviewer are set as the cosine distance of their label vectors. We cluster the reviewers into 5 clusters based on their topic vectors using spectral clustering. We set the constraints such that each paper matches with at least 3 reviewers and every reviewer is allocated at least 1 paper, while no reviewer is allocated more than 10 papers.

Despite the constraints, D-WBM finds a diverse matching with PoD of 0.92. GD-WBM provides an average entropy gain of 1.60 but pays a higher price of diversity as shown in Table 3.2. To delve deeper into the results, we take the example of 48^{th} paper titled "Towards musical query-by-semantic-description using the CAL500 data set" from the dataset. This paper is labeled as related to Topics T8 (Multimedia IR), T16 (Language models) and T3 (Other machine learning). WBM matches it to three reviewers with IDs 43, 34, and 158[2]. If we analyze their topic vectors, we find that all of them have the Language Models (T16) topic common between them. Not surprisingly, they are all found by our clustering method to be in the same cluster, as shown in Fig. 3.3.

[2]More information on the reviewers is available here: `http://sifaka.cs.uiuc.edu/ir/data/review.html`

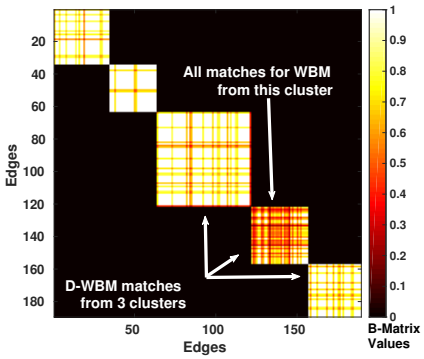

Figure 3.3: Block Diagonal B-Matrix for Paper 48. We notice that WBM selects all matches from a single cluster.

On the other hand, diverse matching provides a match with three reviewers (IDs 131, 153 and 158) from three different clusters. Reviewer 131 provides a balance between IR and Language Model topics but also works on User Studies. Reviewer 153 works on many topics relevant to the paper — Multimedia IR (T8), ML (T2, T3) and Text Categorization (T1). In D-WBM's reviewer set, Reviewers 153 and 131 have no common aspect between them while Reviewer 158 shares interests with both. Having a set of reviewers who are similar to the query paper but complementary in skillsets may provide a well-rounded review with good coverage of different viewpoints. GD-WBM also finds a match which has higher entropy (three different clusters) than WBM.

Scholarly Paper Recommendation Dataset: To further test our method on matching reviewers with papers, we use the Scholarly Paper Recommendation dataset provided by Sugiyama *et al.* [248], which contains 50 researchers and 100,351 candidate papers from proceedings in the ACM Digital Library.

We select all papers from KDD 2000 to KDD 2010 from this dataset (a total of 1184

papers) and find three reviewers for each of them from the given set of 50 reviewers. We calculate edge weights between papers and reviewers as the cosine distance between the TF-IDF vector of the query paper and reviewer's latest paper. No limit of maximum number of papers that a reviewer can review is imposed but each reviewer must be allocated at least one paper. We divide the reviewers into 5 clusters using their TF-IDF vectors with Spectral Clustering. The results show that D-WBM improves on the diversity of all 1184 papers with EG of 4.28 as shown in Table 3.2. The larger EG in this dataset compared to UIUC is because EG decreases as we reduce the upper bound, so the net effect observed in UIUC dataset is also due to choice of bounds. Here, we removed one factor and noticed much larger entropy gain for little loss of efficiency (6%).

3.4.4.4 Effect of Bounds and Problem Size

So far, we have set the cardinality bounds without discussing their effect on the matching results. One would expect that as bounds become tighter, the utility of WBM and D-WBM will suffer due to lesser search space. However, the question we answer here is how it affects the relative performance of the two methods as measured by PoD and EG.

More specifically, we study R^-, as the number of edges in the matching is determined by it. We use UIUC dataset discussed before, where each reviewer must review at least one paper and we cluster the reviewers into 10 groups. Figure 3.4 shows that the PoD is consistently high irrespective of the number of reviewers matched to every paper. The PoD initially decreases as more clusters contribute to diversity but then slowly increases to 1 as the problem becomes more and more constrained. Obviously, when $R^- = M$, there is only one matching possible and both WBM and D-WBM have the same utility.

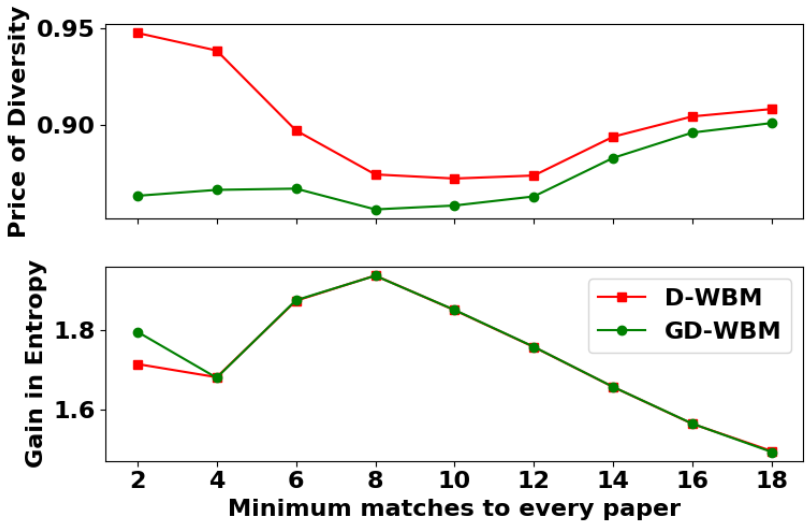

Figure 3.4: Change of PoD and EG with increasing R^-.

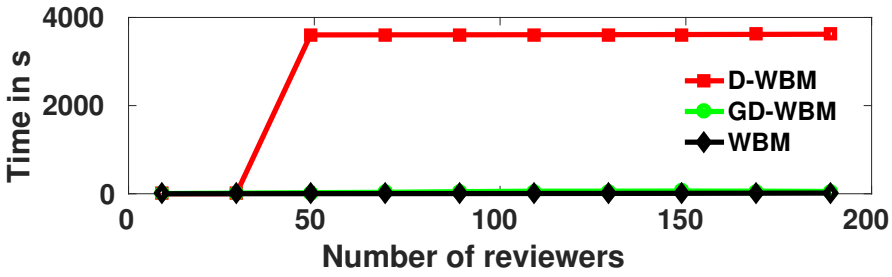

Figure 3.5: Runtime comparison as problem size increases.

EG in general increases when R^- is less than the number of clusters as new clusters can contribute to diversity. Among other bounds, setting R^+ to any value has no effect on optimization. Increasing L^+ allows WBM to overuse few good nodes, who might have low edge weights with everyone. Hence, WBM's entropy suffers significantly.

Finally, we discuss the effect of problem size on the performance of WBM, D-WBM, and GD-WBM. We use the UIUC dataset with $R^- = 3, L^- = 1$ and increase the number of reviewers available to review the papers. Figure 3.5 shows the relative time performance of the three methods. We can see that WBM and GD-WBM take much less

time than D-WBM's MIQP solver. The latter time is capped at one hour and the best solution at that point is used for analysis.

3.4.5 Assumptions and Limitations for Research Task 1:

Below, we list the major assumptions of our work:

1. Our first assumption is that all the nodes in the bipartite graph are known. This means before the allocation of workers to tasks (or reviewers to papers), we know all the available tasks and all the available workers.

2. Our second assumption is that all the edges and edge weights in the bipartite graph are known. This means before the allocation of workers to tasks, we know which task is available to be performed by which worker and the expertise (measured by edge weight) of the worker for the task.

3. We assume that the problem is feasible. The knapsack and cover constraints can be in such a way that the problem becomes infeasible. For instance if all reviewers decide to not review more than three papers but the total number of reviewers required is larger, then the algorithms will not find a feasible solution. We assume that the problem is feasible and allocations can be made.

The major limitation of our work is five-fold. First, it requires a static graph, where all nodes, edges and edge weights are deterministically known. In many practical team formation scenarios, workers may arrive online, one at a time. People may exhibit preferences on the tasks they want to do and may change their preferences dynamically. These

changes will lead to a change in the graph structure and require re-running the algorithm after every change, which may not be practical. Second, the algorithm assumes that each person belongs to a single cluster (or country). However, some people have multiple cluster membership. This means a person can have dual citizenship and they may impart diversity based on both countries. The third limitation of our method is that it does not provide the flexibility to change how much diversity is needed for a given application. We address this limitation in the next research task by providing a new objective function, which has a λ term acting as the diversity knob. If you increase λ, the algorithm provides a more diverse solution. The fourth limitation is the amount of space needed to store the MIQP program. For a medium-sized graph of 500 nodes on each size, the number of terms in the objective function is more than 10 Billion. Storing and solving this large problem is non-trivial with limited computation and storage resources. In the next section, we provide an improvement of our current method which leads to lesser storage. Finally, the D-WBM and GD-WBM's limitation is its lack of guarantee of reaching the optimal solution. For small problems, the algorithm often reaches the optimal solution, as the lower and upper bound estimates converge. However, for larger problems, it often does not. Due to lack of any guarantee, it is difficult to assess how far one is from the optimal solution. In the next section, we address this limitation by providing guarantees on reaching the optimal solution for the diverse matching problem.

3.4.6 Concluding Remarks of Research Task 1

In this research task, we presented a quantitative approach to balancing diversity and efficiency in a generalization of bipartite matching where agents on one side of the mar-

ket can be matched to sets of agents or items on the other. We proposed a quadratic programming-based approach to solving a supermodular minimization problem that balances diversity and total weight of the solution. We proposed the price of diversity (PoD), which measures efficiency loss due to enforcing diversity, and gave worst-case theoretical bounds on that metric. Finally, we validated our methods on three real-world datasets, and showed that the price of diversity is quite good in practice.

However, we found three main areas of improvement in the quadratic formulation for offline diverse matching. Firstly, the MIQP solvers do not provide any theoretical guarantees of reaching the optimal solution. Secondly, the method is not polynomial time. In the next research task, we propose a new algorithm which provides a pseudo-polynomial time algorithm and is guaranteed to reach the optimal solution. Lastly, we modify our objective function to better reflect the trade-off between diversity and efficiency by combining efficiency maximizing WBM and entropy maximizing D-WBM.

3.5 Research Task 2: Negative Cycle Detection for Diverse Matching

Collaborative work often benefits from having teams or organizations with heterogeneous members. In this work, we identify theoretically-tractable segments of the diverse b-matching problem space and propose new algorithms that construct provably-optimal diverse b-matchings in polynomial time. We then compare directly, on real-world datasets, against the state-of-the-art, quadratic-programming-based approach to solving diverse b-matching problems and show that our method outperforms it in both speed and (anytime) solution quality.

C	$\triangleq$	Set of m countries
X	$\triangleq$	Set of n people
T	$\triangleq$	Set of t teams
d_i	$\triangleq$	Number of people needed in team T_i
$u_{i,j}$	$\triangleq$	Utility of assigning each person from country C_j to the team T_i
$c_{i,j}$	$\triangleq$	Number of people assigned from country C_j to team T_i
U	$\triangleq$	Total utility of an assignment
D	$\triangleq$	Total diversity of an assignment
λ	$\triangleq$	Relative importance of diversity compared to utility
λ_1, λ_2	$\triangleq$	Positive integers where $\lambda = \lambda_1/\lambda_1$
$(I_j^k,\ O_j^k)$	$\triangleq$	Input and Output port in switch T_k of the auxiliary graph G' corresponding to team k and country C_j
$w(e_{i,j}^k)$	$\triangleq$	weight of edge in auxiliary graph G' from country i to country j within switch for team T_k
U	$\triangleq$	Weight of the minimum weighted b-matching

Table 3.4: Table of notation for Research Task 2.

3.5.1 Preliminaries

In this task, we use slightly different notations compared to the previous research task. Below, we define the preliminaries which we will use for the diverse matching problem, where workers are to be matched to teams and each team wants workers belonging to a diverse set of countries. In our problem, we are given a set of countries $\mathcal{C} = \{C_1, \ldots, C_m\}$ and each country C_i has $|C_i|$ people inside it. The set of people is denoted by $X = \{x_1, \ldots, x_n\}$. We wish to form a set of teams $\{T_1, \ldots, T_t\}$ of the people where each team T_i has a demand d_i specifying the number of people that needs to be assigned to it. Each person can be assigned to exactly one team. The utility of assigning each person from country C_j to the team T_i is denoted by $u_{i,j}$. We assume all utilities are integers. The number of people assigned to team T_i from country C_j is $c_{i,j}$. The total utility of an assignment is $U = \sum_{i=1}^{t} \sum_{j=1}^{m} u_{i,j} \cdot c_{i,j}$. The diversity of an assignment is denoted by D and is equal to $\sum_{i=1}^{t} \sum_{j=1}^{m} c_{i,j}^2$. The goal is to minimize the objective function which is equal to $\lambda \cdot D + U$, where $\lambda > 0$ is a constant. We assume λ is a rational number and $\lambda = \frac{\lambda_1}{\lambda_2}$ where $\lambda_1, \lambda_2 \in \mathcal{Z}^+$. We also assume λ_1, λ_2 are constants. Minimizing $\lambda \cdot D + U$ is equivalent to minimizing $\lambda_1 \cdot D + \lambda_2 \cdot U$ and we will focus on this new objective function.

In §3.5.4, we show how to generalize this framework to solve the problem for the case that the utilities of assigning people from the same country to a team could be different. We call that diverse bipartite b-matching with general weights.

Matrix Representation: In this representation, each column corresponds to a country, and each row corresponds to a team. Entry $c_{i,j}$ shows the number of people from the country C_j assigned to the team T_i. Team T_0 is a dummy node, and $c_{0,i}$ shows

the number of people from country C_i which are not assigned to any team. An example is shown in Fig. 3.6.

Matching Representation: Here, a bipartite graph is given. The nodes on the left show different countries and people belonging to those countries, and the nodes on the right represent the teams. The assignment of people to the teams form a b-matching. In Figure 3.7, two possible assignments are shown with different colors. The red arrows in Figure 3.7 show a local exchange which is explained in the following.

Local Exchange: A local exchange happens when a group of teams decide to transfer one or more workers to each other. The exchange is done in a way that the initial demands of all the teams are fulfilled. Red arrows in Figure 3.6 show a local exchange using a matrix representation. In this exchange, one person from C_2 is moved from T_3 to T_1. Two persons from C_1 are moved. One is moved from T_1 to T_2, and the other one is moved from T_2 to T_3. The set of edges of a local exchange in a matrix representation is called a cycle. The sources of a cycle are the cells without any input edges, and the sinks are the cells without any output edges. In Figure 3.6, the nodes corresponding to $c_{3,2}$ and $c_{1,1}$ are source nodes, and the nodes corresponding to $c_{1,2}$ and $c_{3,1}$ are sink nodes.

Figure 3.7 shows the same local exchange operation using a matching representation. In this figure, the black matching shows the initial assignment, and the blue matching shows the assignment after the exchange operation is done.

Gain of a local exchange: It is possible that a local exchange may improve the net objective function. To find out, we first calculate the marginal gain from a given exchange operation which is the difference between the final and initial objective values. For example, gain of the local exchange in Figure 3.6 is $\lambda_1((c_{3,2}-1)^2 - c_{3,2}^2 + (c_{1,2} +$

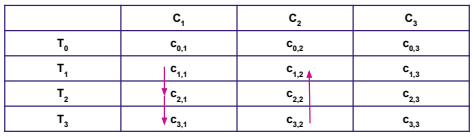

Figure 3.6: Matrix representation of three teams and workers from three countries. Dummy team T_0 accommodates unassigned workers. Red arrows represent a local exchange.

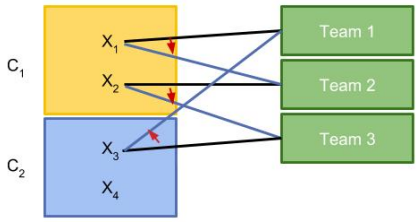

Figure 3.7: Local exchange operation (in matching representation).

$1)^2 - c_{1,2}^2 + (c_{1,1} - 1)^2 - c_{1,1}^2 + (c_{3,1} + 1)^2 - c_{3,1}^2) + \lambda_2(-u_{3,2} + u_{1,2} - u_{1,1} + u_{3,1})$. It can be seen the contribution of the nodes which are not source or sink to the gain of a local exchange is zero. If the net gain is negative, then the local exchange can be considered successful.

3.5.2 Negative-Cycle-Detection-based Algorithm

In this section, we explain our algorithm for finding the optimum assignment. First, we build the following graph G'. For each team T_i, there is a switch in G' with m input ports, and m output ports, where m is the number of countries. There is a dummy team T_0 to accommodate all unassigned workers in the matching. For each pair of input output ports (I_i^k, O_j^k) in switch T_k, there is a directed edge e_{ij}^k from I_i^k to O_j^k, its weight is defined in

the following way:

$$w(e_{ij}^k) = \begin{cases} -2\lambda_1 & \text{if } i = j, i \neq 0 \\ 0 & \text{o.w} \end{cases}$$

The reason behind putting the weight of edges in the first case equal to $-2\lambda_1$ is to force the nodes which are not a source or sink and do not belong to T_0 have zero contribution to gain of a cycle.

For each pair of teams T_i and T_j where $i \neq j$, and for each country c_ℓ, there is a directed edge from output port O_ℓ^i of switch T_i to the input port I_ℓ^j of switch T_j, and weight of this edge equal to:

$$\begin{cases} \lambda_1((c_{j,\ell}+1)^2 - c_{j,\ell}^2 + (c_{i,\ell}-1)^2 - c_{i,\ell}^2) & \\ \qquad +\lambda_2(u_{j,\ell} - u_{i,\ell}) & i, j \neq 0 \\ \lambda_1((c_{j,\ell}+1)^2 - c_{j,\ell}^2) + \lambda_2(u_{j,\ell}) & i = 0 \\ \lambda_1((c_{i,\ell}-1)^2 - c_{i,\ell}^2) + \lambda_2(-u_{i,\ell}) & j = 0 \end{cases}$$

The reason behind separating the cases is that T_0 is a dummy node which shows the number of unassigned people from each country, and its contribution to the objective function must be zero.

Each cycle in this graph is corresponding to a cycle in a matrix representation and local exchanges along them have the same gain. Figure 3.8 shows a cycle which is corresponding to the cycles in Figures 3.6 and 3.7. In this cycle, weight of edges is defined in

the following:

$$w(e_1) = 0$$

$$w(e_2) = \lambda_1((c_{2,1}+1)^2 - c_{2,1}^2 + (c_{1,1}-1)^2 - c_{1,1}^2) +$$

$$\lambda_2(u_{2,1} - u_{1,1})$$

$$w(e_3) = -2\lambda_1$$

$$w(e_4) = \lambda_1((c_{3,1}+1)^2 - c_{3,1}^2 + (c_{2,1}-1)^2 - c_{2,1}^2) +$$

$$\lambda_2(u_{3,1} - u_{2,1})$$

$$w(e_5) = 0$$

$$w(e_6) = \lambda_1((c_{1,2}+1)^2 - c_{1,2}^2 + (c_{3,2}-1)^2 - c_{3,2}^2)$$

$$+ \lambda_2(u_{1,2} - u_{3,2})$$

After constructing the graph, we run Algorithm 3. Algorithm 3 moves people from one team to another within the same country if it detects a negative cycle. The movement of person is always from output port of a team to the input port of another team.

Algorithm 3 gets an initial feasible solution M as input. To find M, we first solve a mixed integer program to find the minimum weight solution, which satisfies the constraints. This solution is also the baseline, as the gain in diversity by diverse matching is with respect to this solution. In Algorithm 3, to detect negative cycles in G', we use a heuristic improvement of Bellman-Ford proposed by Goldberg and Radzik [104].

Algorithm 3: Find optimal diverse b-matching.

Input : Directed weighted graph G', initial feasible b-matching M which satisfies team demands.

Output: Optimal diverse b-matching

while $\exists$ *a negative cycle* $C \in G'$ **do**
    // Perform a local exchange operation along C;
    for $e \in C$ **do**
        // Assume edge e is from output port O_ℓ^i of team T_i to input port I_ℓ^j of another team T_j;
        // Move one person of country c_ℓ from team T_i to team T_j:
        $c_{i,\ell} -= 1$;
        $c_{j,\ell} += 1$;
        Update weight of edges of G' w.r.t to the new values of $c_{i,\ell}$, and $c_{j,\ell}$;
    end
end

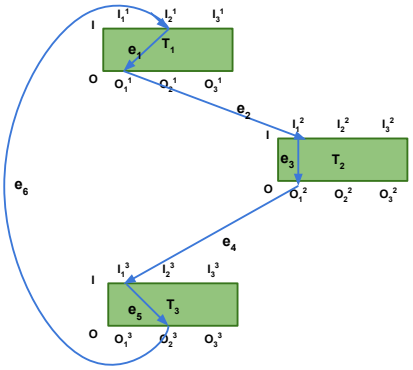

Figure 3.8: Local Exchange in Graph Representation.

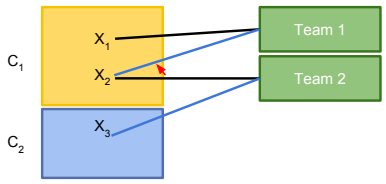

Figure 3.9: Example of a local exchange along an alternating path.

3.5.3 Proof of Optimality

In this section, in Theorem 2, we prove that Algorithm 3 gives the optimal solution for diverse bipartite b-matching problem.

Assume after the algorithm ends, the final assignment is a local optima M, and the optimum solution is M^*. Consider the matching representations of M and M^*. The symmetric difference of M and M^* ($M \oplus M^*$) can be decomposed into a set of alternating cycles and paths of even length. The reason that the length of alternating paths is even is that size of both of the matchings is equal: $|M| = |M^*| = \sum_{i=1}^{t} d_i$.

Each local exchange along an alternating cycle is corresponding to a cycle in a matrix representation. A local exchange along an alternating path is corresponding to a cycle in a matrix representation which includes vertices from the row T_0. For example, Figure 3.9 shows a local exchange along an alternating path in $M \oplus M^*$ where blue edges belong to M^*, and black edges belong to M. Matrix representation corresponding to this local exchange is shown in Figure 3.10.

Before proving Thm. 2, we need the following definitions:

Maximal Cycle: A cycle c in the matrix representation is maximal if its sources (nodes with zero incoming edges) and sinks (nodes with zero outgoing edges) are source

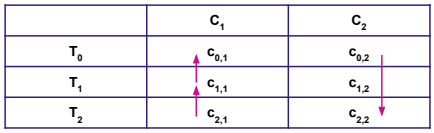

Figure 3.10: Matrix representation corresponding to an alternating path.

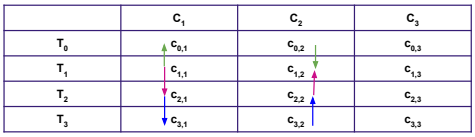

Figure 3.11: Maximal Cycle Decomposition of graph.

and sink w.r.t all the edges in $M \oplus M^*$ as well. For example, consider the cycles in Figure 3.11. Let's call the green cycle c_g, the red cycle c_r, and the blue cycle c_b. The green cycle has two sources $c_{1,1}, c_{0,2}$, and it has two sinks $c_{0,1}, c_{1,2}$. Since there are no edges going out of $c_{1,2}, c_{0,1}$, and no edges going into $c_{0,2}, c_{1,1}$, c_g is a maximal cycle. Cycle c_r is not a maximal cycle since $c_{2,2}$ is a source w.r.t the red edges, but it is not a source w.r.t all edges and a blue edge is going into it. Also $c_{2,1}$ is a sink w.r.t the red edges, but there is a blue edge going out of it. Cycle c_b is not a maximal cycle as well.

Lemma 1. *The set of all the edges of $M \oplus M^*$ in matrix representation can be decomposed into a set of maximal cycles.*

Proof. Consider an arbitrary decomposition of the edges of $M \oplus M^*$ in the matrix representation into a set of cycles $\{c_1, \cdots, c_\ell\}$. If there exists a cycle in $M \oplus M^*$ without any source and sink nodes, it means the gain of this cycle is zero and it could be discarded. If there exists any cycle c_i which is not maximal, then there exists another cycle c_j which makes c_i not to be maximal. For example in Figure 3.11, c_r is not maximal because of c_b.

In this case, union c_i and c_j, and make $c_i \cup c_j$ a single cycle in the decomposition. At the end, all the edges in $M \oplus M^*$ will be decomposed into a set of maximal cycles. Let's call the set of maximal cycles $\{c'_1, \cdots, c'_j\}$. □

For example, in Figure 3.11, decomposition $\{c_g, c_r \cup c_b\}$ is a maximal cycle decomposition.

Now we are ready to prove the following theorem:

Theorem 2. *Algorithm 3 finds the global optimum for the diverse b-matching problem.*

Proof. Let $f(M)$ show the value of the objective function for the assignment M. $f(M^*) - f(M) < 0$ therefore:

$$f(M^*) - f(M) = gain(c'_{1,1}) + gain(c'_{2,2}) + \cdots + gain(c'_{j,j}) < 0$$

Where c'_k $(1 \leq k \leq j)$ is the k^{th} cycle in the maximal cycle decomposition, and $c'_{k,k}$ is applying the local exchange of the cycle c'_k at step k. The initial step is the assignment M. Since $f(M^*) - f(M) < 0$, there must be a maximal cycle c'_i such that $gain(c'_{i,i}) < 0$. We wish to show $gain(c'_{i,1}) < 0$. Which implies starting from the initial assignment M, a local exchange can be done with a negative gain, and M is not a local optima which is a contradiction.

Consider $c'_{i,i}$ in a matrix representation. There are four types of vertices in $c'_{i,i}$:

- Vertices in the form of $c_{0,i}$ in Figure 3.6 where $1 \leq i \leq m$. This vertices have contribution zero to both $gain(c'_{i,i})$ and $gain(c'_{i,1})$.
- Vertices that are not sink or source, like $c_{2,1}$ in Figure 3.6. It could be seen that contribution of these nodes to both $gain(c'_{i,i})$ and $gain(c'_{i,1})$ is zero.

- Sink vertices: Consider an arbitrary sink node v in $c'_{i,i}$. Assume the value of this node at the beginning of step i is v_i. The contribution of v to $gain(c'_{i,i})$ is positive and is equal to $\lambda_1((v_i+1)^2-v_i^2)+\lambda_2 u$. Since v is a sink node and there are no edges out of v, $v_i \geq v_1$. Therefore, $\lambda_1((v_i+1)^2 - v_i^2) + \lambda_2 u \geq \lambda_1((v_1+1)^2 - v_1^2) + \lambda_2 u$. As a result, the contribution of v to $gain(c'_{i,1})$ is upper bounded by its contribution to $gain(c'_{i,i})$.

- Source vertices: Consider an arbitrary source node v in $c'_{i,i}$. Assume the value of this node at the beginning of step i is v_i. The contribution of v to $gain(c'_{i,i})$ is $\lambda_1((v_i-1)^2 - v_i^2) - \lambda_2 u$. v is a source node and therefore $v_1 \geq v_i$. As a result, $\lambda_1((v_i-1)^2 - v_i^2) - \lambda_2 u \geq \lambda_1((v_1-1)^2 - v_1^2) - \lambda_2 u$.

At the end, contribution of all the vertices to $gain(c'_{i,1})$ is upper bounded by their contribution to $gain(c'_{i,i})$. Therefore if $gain(c'_{i,i}) < 0$, then $gain(c'_{i,1}) < 0$. □

Theorem 3. *The running time of the algorithm is $\mathcal{O}((\lambda_1 \cdot n^2 + \lambda_2 U) \cdot m^2 \cdot t^2(m+t))$, where U is the weight of the minimum weighted b-matching w.r.t the utility weights.*

In order to prove this theorem, first we show the following lemmas hold.

Lemma 2. *The number of iterations of our algorithm is at most $(\lambda_1 \cdot n^2 + \lambda_2 U)$.*

Proof. The initial state of the algorithm is a feasible b-matching with weight U. Diversity of any matching is at most n^2. Therefore, the maximum value of the objective function is at most $\lambda_1 \cdot n^2 + \lambda_2 U$. At each iteration, we find a negative weight cycle and since all the weights are integers its weight can be at most -1. Therefore the objective function decreases by at least 1 at each step, and since the value of the objective function is always positive, the number of iterations is at most $\lambda_1 \cdot n^2 + \lambda_2 U$. □

Lemma 3. *The complexity of each iteration of the algorithm is* $\mathcal{O}(m^2 \cdot t^2(m+t))$.

Proof. At each iteration, we use a negative cycle detection algorithm with running time $\mathcal{O}(V \cdot E)$ to detect a negative cycle. The number of nodes in the graph is $2m \cdot (t+1)$, since there are $t+1$ switches in the graph and each switch has exactly $2m$ ports and each port is a node of the graph. The number of edges incident on each port is exactly $m+t$. Therefore, the total number of edges is $\mathcal{O}(m \cdot t(m+t))$. Hence, the complexity of each iteration is $\mathcal{O}(m^2 \cdot t^2(m+t))$. □

By putting Lemma 2 and Lemma 3 together, Theorem 3 can be proved.

3.5.4 Bipartite b-Matching with Different Weights for each Worker

In order to extend our framework to solve the case where utility of assigning people from the same country to a team can be different, we make the following modifications to the algorithm we mentioned in Section 3.5.2. First, in each switch instead of putting input and output ports for each country, we put input and output ports for each person. Inside each switch $T_0, T_1 \cdots, T_t$, there is a complete bipartite graph from input ports to the output ports. Consider an arbitrary switch T_ℓ where $1 \leq \ell \leq t$. $w^\ell_{i,j}$ shows the weight of input port x^ℓ_i to the output port x^ℓ_j in this switch. Then:

$$w^\ell_{i,j} = \begin{cases} -2\lambda_1 & \exists C_k \in \mathcal{C} : x_i, x_j \in C_k \\ 0 & \text{o.w} \end{cases}$$

This weight function ensures if x_i is assigned to T_ℓ and x_j is moved out of T_ℓ, the diversity of T_ℓ does not change if x_i and x_j are from the same country, and its diversity changes otherwise. The weight of inter-switch edges are computed similar to what we did in

§3.5.2. Consider an edge from output port x_ℓ^i of switch T_i to the input port x_ℓ^j of switch T_j. Assume x_ℓ is from country C_k. The weight of this edge is equal to the change in the objective function by moving one person from C_k out of T_i, and adding that person to T_j. The following theorem holds similar to the way that Theorem 3 was proved.

Theorem 4. *The running time of the algorithm for general weights is* $\mathcal{O}((\lambda_1 \cdot n^2 + \lambda_2 U) \cdot n^2 \cdot t^2(n+t))$*, where* U *is the weight of the minimum weighted b-matching w.r.t the utilities.*

3.5.5 Results and Discussion

To demonstrate the efficacy of the proposed method, we apply it to two domains: matching movies to users and matching reviewers to papers. First, we show the trade-off front between diversity and total weight of matching for reviewer assignment problem. Next, we compare our algorithm with existing methods in the literature and show that it outperforms them in terms of time taken to converge.

3.5.5.1 Application to Reviewer Assignment

We now present an application of diverse matching to automatically determine the most appropriate reviewers for a manuscript by also ensuring that reviewers are different from each other. We use the multi-aspect review assignment evaluation dataset [143], a benchmark dataset from UIUC. It contains 73 papers accepted by SIGIR 2007, and 189 prospective reviewers who had published in the main information retrieval conferences. The dataset provides 25 major topics and for each paper in the set, an expert provided 25-dimensional label on that paper based on a set of defined topics. Similarly for the 189

reviewers, a 25-dimensional expertise representation is provided.

To set up the graph, we first cluster the reviewers into 5 clusters based on their topic vectors using spectral clustering. To calculate the relevance of each cluster for any paper, we take the average cosine similarity of label vectors of reviewers in that cluster and the paper. We set the constraints such that each paper matches with at least 3 reviewers and no reviewer is allocated more than 2 papers.

We first find the maximum weight matching for this problem. The resultant matching is found to be non-diverse. In fact, all 73 papers are allocated three reviewers who are all from the same cluster. This gives the resultant matching zero diversity, as measured by Shannon entropy. Next, using the bi-objective formulation, we show that by varying λ, we can obtain the trade-off between utility and diversity. Figure 3.12 shows the trade-off front between average Shannon entropy and total weight of the matching for different values of λ. For this problem, once λ is greater than 0.26, all matchings are maximum diversity matching and they result into the same matching allocation.

The trade-off front allows us to explore how diversity affects the total weight of the matching for any given domain. For instance, in Fig. 3.12, by marginally increasing λ above 0, we see large gain in entropy with little loss in total weight of the matching. In the subsequent sections, we set $\lambda = 1$ fixed.

3.5.5.2 Application to MovieLens Data

In this section, we compare our algorithm with MIQP approach with increasing size of the graph. This example considers matching movies to users, while ensuring that the movies contain diverse genres. We use a subset of the MovieLens 1M dataset [117], which

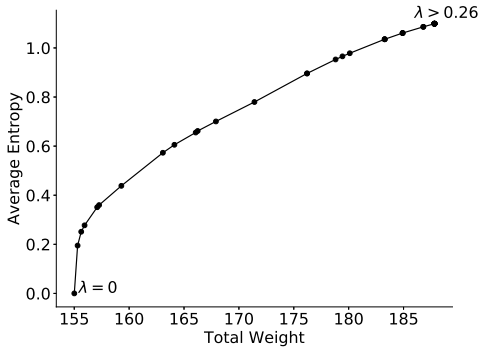

Figure 3.12: Trade-off front between utility and entropy.

includes one million ratings by 6,040 users for 3,900 items. This dataset contains both users' movie ratings between 1 and 5 and genre categories for each movie (*e.g.*, comedy, romance, and action). We first train a standard collaborative recommender system [40] to obtain ratings for all movies by every user. We cluster the movies into 5 clusters using their vector of 18 genres using spectral clustering, so that each movie gets a unique cluster label.

We solve the matching problem for 1500 movies and make three recommendations for each user. The number of users are increased in steps of 50 to compare the timing performance of our approach to MIQP. For MIQP, we set a maximum run time of two hours (7200 seconds), at which we report the current best MIQP solution. Table 3.5 shows that for all cases with number of movies greater than 75, MIQP does not converge within two hours, while our method finds the optimal solution in lesser time. Interestingly, MIQP current solution is found to be the same as the optimum solution showing that MIQP is able to search the solution but not able to prove it.

Number of movies	MIQP Time (s)	Our Method Time (s)
50	133	33
75	7200	98
100	7200	247
125	7200	663
150	7200	1069
175	7200	2581
200	7200	2316
250	7200	4609

Table 3.5: Running time comparison of MIQP and our method for MovieLens dataset.

3.5.6 Assumptions and Limitations for Research Task 2:

Below, we list the major assumptions of our work:

1. Our first assumption is that all the nodes in the bipartite graph are known. This means before the allocation of workers to tasks, we know all the available tasks and all the available workers.

2. Our second assumption is that all the edges and edge weights in the bipartite graph are known. This means before the allocation of workers to tasks, we know which task is available to be performed by which worker and the expertise (measured by edge weight) of the worker for the task.

3. We assume that each worker belongs to a cluster and diversity is defined as coverage over all clusters.

4. To initiate the negative cycle detection algorithm, we provide a feasible initial solution. We assume that such a feasible solution exists and can be computed.

The major limitation of our work is three-fold. First, it requires a static graph, where all nodes, edges and edge weights are deterministically known. In many practical team formation scenarios, workers may arrive online, one at a time. People may exhibit preferences on the tasks they want to do and may change their preferences dynamically. These changes will lead to a change in the graph structure and require re-running the algorithm after every change, which may not be practical. Second, the algorithm assumes that each person belongs to a single cluster (or country). However, some people have multiple cluster membership. This means a person can have dual citizenship and they may impart diversity based on both countries. The third limitation of our work is the need to run a negative cycle detection algorithm repeatedly. After every transfer of worker, we run the algorithm again to find a new negative cycle. Future work can investigate methods to reduce the number of times this computationally expensive step is needed. One possible way is to find all non-overlapping negative cycles and transfer people simultaneously. However, this will require developing efficient algorithms to find such cycles. Future research can study ways to improve the efficiency of finding negative cycles on auxiliary graphs.

3.5.7 Concluding Remarks of Research Task 2

In this task, we developed an algorithm (and extensions) that is guaranteed to find the global optima of the diverse, weighted, bipartite b-matching problem. Our method is also faster than the D-WBM approach in previous task, on the real-world benchmark test cases. However, many open challenges still exist. For offline matching, it is important to reduce the computational cost for large graphs. Future work in this field can also focus on diverse matching with diversity of a set defined using Determinantal Point Process (DPP) kernels [162], which do not require explicit clustering.

While offline matching is important when a pool of workers is present and tasks need to be allocated within them, in many scenarios the workers arrive sequentially. To address diverse matching for situations where edges arrive sequentially, we will focus on diverse matching for online problems in the next task.

3.6 Research Task 3: Online Diverse Matching

In this task, we present a method to form diverse teams from people arriving sequentially over time. We define diversity using a submodular function and adopt an online algorithm to solve monotone submodular maximization problem with multiple capacity constraints. This allows us to balance both how diverse the team is and how well it can perform the task at hand. Using crowd experiments, we show that, in practice, the algorithm performs much better than theoretical bounds. We also show how simulations can help set key parameters for online matching and provide insights into quantifying the need of diversity. Our method has many applications for collaborative work like team formation,

assignment of workers to tasks in crowdsourcing and reviewer allocation to journal papers arriving sequentially.

3.6.1 Diversity in Matching

This section introduces some of the more detailed mathematical notation needed to properly describe our algorithm for team formation in the next section. We flesh out in more precise detail how diversity is modeled and calculated via a submodular function and how this relates to matching people to teams.

We model the overall problem as maximizing a monotone submodular function over b-matchings in a bipartite graph $G = (P, T, E)$, where P is a set of vertices (*e.g.*, people) that arrive online, T is a set of vertices (e.g., teams) known a priori, and where no vertex i (task or people) is incident to more than $b(i)$ edges in a proposed matching (*i.e.*, we cannot assign a person to more than b teams at once). Even the offline version of this problem is NP-Hard, so we focus on approximate submodular maximization and instead bound how close we can get to the optimal solution. To incorporate diversity, we consider a scenario where left-side nodes (*e.g.*, the people) are divided into K groups (as shown in Fig. 3.13). We want a matching which allocates each node on the right side (a team) to nodes from different clusters on left side (people). A set of edges is considered diverse if it connects left side nodes (people) from different clusters. For example, in Fig. 3.13, matching team $t1$ to person $p1$ and $p2$ is a non-diverse matching (as both $p1$ and $p2$ come from same color block), while matching it to $p1$ and $p3$ is considered diverse.

In this work, we use a square-root-based diversity reward function which balances the number of nodes (e.g., people) selected from different clusters, adapted from the work

N	$\triangleq$	Number of teams to be formed
M	$\triangleq$	Maximum number of people that can arrive sequentially
P	$\triangleq$	Set of vertices that arrive online (*e.g.*people)
T	$\triangleq$	Set of all vertices known apriori (*e.g.*teams)
E	$\triangleq$	Set of all edges
$b(i)$	$\triangleq$	Maximum edges that can be matched to each node (team or people)
L^+	$\triangleq$	Maximum teams that can be matched to each worker
R^+	$\triangleq$	Maximum workers that can be matched to each team
K	$\triangleq$	Number of clusters into which online side of nodes are partitioned
S_j	$\triangleq$	Subset of edges in a matching that are incident to vertex j
$w_{i,j}$	$\triangleq$	Utility of worker i for task j
B	$\triangleq$	Total budget of a firm hiring workers
c_i^S	$\triangleq$	Cost of interviewing a person i
$c_{i,j}^B$	$\triangleq$	Payment to a worker i for team j after acceptance
d	$\triangleq$	Total number of knapsack constraints
$c_{i,j}$	$\triangleq$	Cardinality cost of an edge from worker i to team j, which equals one
S	$\triangleq$	A feasible task allocation
$y_{k,j}$	$\triangleq$	Number of people from cluster k, allocated to team j
df_{R^-}	$\triangleq$	R^- highest marginal gain among all clusters

Table 3.6: Table of notation for Research Task 3.

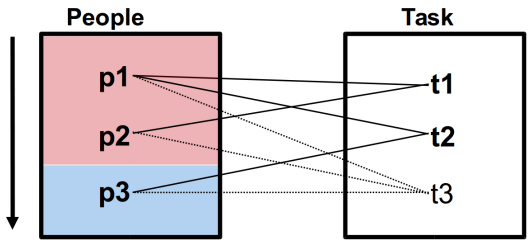

Figure 3.13: Bipartite graph of people arriving online and tasks requiring a team of two each. People belong to two groups here (red and blue). Task $t1$ is matched to two people from the same group while $t2$ is matched to a diverse set of people. Task $t3$ remains unmatched so far.

of [172] on multi-document summarization. We first define some notations. $S_j \subseteq E$ is the subset of edges in a proposed matching that are also incident to vertex j. Assuming people belong to K clusters—*e.g.*, of skillsets or levels of experience — $P_k \subseteq P, k \in [K]$ is a partition of all people P (*i.e.*, $\cup_k P_k = P$ and $P_k \cap P_{k'} = \emptyset$ for all $k \neq k'$). This means that each edge can only belong to one cluster. We also define $w_{i,j}$ as the utility of worker i to do task j. In our context, for a specific task $j \in T$, we define a function $f_j : E \to \mathbb{R}$ which rewards diversity as follows:

$$f_j(S_j) = \sum_{k=1}^{K} \sqrt{\sum_{\{i \mid i \in P_k \wedge (i,j) \in S_j\}} w_{i,j}} \tag{3.8}$$

The part within the square root function controls the quality such that a higher weight $w_{i,j}$ implies the person i offers higher utility (better expertise or higher quality) for the job j. On the other hand, summation of the square root function reduces the marginal gain from adding nodes from the same cluster. Hence, it promotes diversity by preferring people from groups that have not been well represented in the teams so far. The specific form of the diversity function (*i.e.*, the square root form in Eq. 3.8) is not central to the

main contributions of the research task; in practice any preferred submodular function can be substituted instead without significantly affecting any of our main results.

Maximizing $\sum_{j \in T} f_j(S_j)$ over all legal matchings S allows us to solve the offline diverse matching problem. To solve the offline problem, submodular function maximization techniques [30] can be used; however, this assumes that we know exactly all of the people who will be available now and in the future. In the next section, we define the online variant of this problem where we do not assume to know exactly which people will arrive in the future and perform matching "on-the-fly," which more accurately mirrors real-world team formation.

3.6.2 Online Team Formation

In our online model for team formation/assignment, we model people and teams with a bipartite graph $G(P, T, E)$ where an edge $e = (i, j) \in E$ represents whether a person $i \in P$ can perform task or join a team $j \in T$. Teams are represented as the right side of bipartite graph and people are considered on the left side. There is a firm with a limited budget B and a set of N heterogeneous teams T that need to formed. People arrive one at a time from a large pool P. Each person $i \in P$ has a fixed cost c_i^{S} which is the cost of interviewing or screening the person, during which we learn their attributes (*e.g.*, demographic information, skillset, *etc.*). After the interview/screener, the firm must either assign the person to one or more teams, or reject the person. When a person is accepted for team j, she receives a payment/salary/bonus of $c_{i,j}^{\mathrm{B}}$. Note that while we mentioned using $b(i)$ to refer to the upper bound for any node i, to differentiate between the upper capacity of teams and people on the two side of graphs, we use notations R^+ and L^+ also.

Each team has an upper budget R^+ of the maximum number of workers it needs. Each person has an upper budget L^+ of the maximum number of tasks/teams she is willing to simultaneously participate in. Every time a person is interviewed/screened, the set of edges from the person to all teams is considered to "arrive."

Each person i has a weight $w_{i,j}$ representing the local utility (*i.e.*, fit, value, *etc.*) derived by the firm after matching her to j (we assume that after team formation, the person actually works). We use M to denote the maximum number of people who can arrive, which is assumed to be known by the firm; typically, M is determined by the firm's budget and screening cost c_i^{S}.

With this setup, our problem can now be formulated as an online submodular maximization problem with N knapsack constraints — the N tasks' upper bounds R^+.

3.6.2.1 Overview of our Streaming Algorithm

To perform online team formation, we treat people as a continuous *stream*, and build upon past approaches to streaming algorithms to solve online diverse matching. Specifically, our objective function is monotonic submodular with an upper bound on the cardinality of people and teams. A recently proposed algorithms by [279] attempts to solve the problem of online submodular maximization with d knapsack constraints, for $d \in \mathbb{N}$ (fully described as Algorithm 4 of [281]). This algorithm estimates optima for the offline problem based on items observed up to any time step and then accepts or rejects edges based on feasibility and marginal gain being above a cut-off value. An optimum is estimated either using maximum possible marginal gain over all edges, or the current maximum marginal gain.

It is easy to assume that their algorithm can be directly applied to the team formation problem. However, if one looks carefully, they will notice that it cannot be practically applied to real-world matching due to two reasons. First, it maintains multiple *separate assignment solutions* and, when people arrive, they are accepted or rejected for each list separately. An arriving edge can be accepted by multiple lists and rejected by others. Practically, this would mean that when a person arrives at a firm, he or she is possibly allocated to several teams and rejected by others. The person does their allocated job for all the teams or tasks they are accepted for and the firm maintains multiple possible allocations simultaneously. Using [281], after completion of the online allocation phase, the firm would then "select" the allocation list that has with maximum utility. This would mean that many people previously allocated to (and already working on) teams would then be rejected. If a person has completed the task already, then their output gets wasted.

Second, their algorithm has only capacity constraints, implying that in many situations, teams may receive fewer people than its upper bounds (due to strict filtering). This can be problematic in practical scenarios, where tasks often require atleast a minimum number of people and have upper capacity too — *i.e.*, have both coverage and knapsack constraints.

In this work, we address these two issues to leverage the algorithm of [281] for practical team formation. We use Algorithm 4 [281] for submodular maximization with d-knapsack constraints, where an approximation of optima (OPT) is known. When only capacity constraints exist for each task, we have $d = N$ constraints. In this algorithm, $c_{i,j}$ is the cost of admitting a worker i, which is 1 when only capacity constraints exist. In this case, the maximum capacity of a team b equals R^{+}. Running this algorithm requires

an α-approximation of the global optimum for the offline case, $\alpha \in (0, 1]$. As discussed in [281], this algorithm provides a $\frac{\alpha}{1+2d}$-approximation guarantee of the optimal solution, where d is the number of knapsacks and α is the approximation factor upto which we can estimate the optima OPT.

Algorithm 4: Online Diverse Matching

Input: v such that $\alpha OPT \leq v \leq OPT$, $\alpha \in (0, 1]$

Output: A feasible task allocation $S \subseteq E$

$S \leftarrow \emptyset$

for $i \leftarrow 1$ ***to M*** **do**

  if $c_{i,j} \geq \frac{b}{2}$ *and* $\frac{f(\{i\})}{c_{i,j}} \geq \frac{2v}{b(1+2d)}$, *for any* $j \in [d]$ **then**

    $S = \{i\}$; **return** S

  if $\sum_{l \in S \cup i} c_{l,j} \leq b$ *and* $\frac{f(i)}{c_{i,j}} \geq \frac{2v}{b(1+2d)}$, $\forall j \in [d]$ **then**

    $S = S \cup \{i\}$

return S

To adapt this algorithm to online team formation, we solve the problem in three steps. First, we define a convex optimization problem and solve it to estimate an upper bound on OPT. Second, instead of individual edges (items) arriving online, we have a batch of edges (corresponding to all teams a person could join) arriving online. We sort these edges with respect to marginal utility and send them in that order. By prioritizing tasks more suited to the skillset of a person, we improve the performance of our algorithm by giving strictly better results than random task order. Finally, we discuss setting α using marginal gains for clusters to guarantee that we can satisfy lower bounds too (given un-

limited arrival of people). Note that in Algorithm 4 we have not explicitly mentioned the case with capacity constraints on people (when each worker cannot do more than L^+ jobs) or monetary budget constraints (when maximum budget B is given for team formation), but adding these constraints is straightforward and does not change the algorithm. To add any additional constraints like budget or person capacity, we only need to define the individual cost incurred in selecting the corresponding node and the total budget allowed. For instance, considering the monetary case would mean cost $c_{i,j}$ in Alg. 4 equals $c^{\mathrm{B}}_{i,j}$ for the budget constraints and upper bound b equals B. We do not model the screening cost c^{S}_i in accepting or rejecting a worker.

3.6.2.2 Estimating the Optimum: Finding Maximum Number of People from each Cluster

To estimate the optimum for the offline problem, we assume an unlimited stream of people exists, without knowing the number of people arriving from each cluster or their order. We make two assumptions. First, we assume that all people from the same cluster provide similar utility for any given team/task and, second, we assume that people are willing to participate in all tasks. With these assumptions, we can formulate the diversity maximization problem for all teams by summing up submodular gains across each team and each cluster from Eq. 3.8. Let $y_{k,j}$ be the number of people from cluster k matched to team j. Let $w_{k,j}$ be utility of a worker from cluster k matched to team j. In this problem, we only consider R^+ upper cardinality constraints on the maximum number of people for a given team. Hence we define the following problem:

$$\max_{y} \sum_{j=1}^{N} \sum_{k=1}^{K} \sqrt{w_{k,j} y_{k,j}} \quad \text{s.t.} \quad \sum_{k=1}^{K} y_{k,j} \leq R_j^+ \ \forall j \in [N] \tag{3.9}$$

This is a concave maximization problem with linear constraints, and can be solved using a convex solver for real valued y and optimum value OPT^*. A mixed-integer convex solver can also be used to obtain the true OPT [180]; however, such solvers are still in their nascency and, as we discuss later, the real-valued relaxation is sufficient for our case.

Solving Eq. 3.9 with real valued y yields OPT^*, which satisfies $\alpha OPT \leq v \leq OPT \leq OPT^*$. Solving this problem essentially estimates how many people from each cluster we should expect in an optimal solution and *not* the allocation of individual people (as people are exchangeable within a cluster). We use OPT^* in place of OPT to filter edges in Algorithm 4.

The optimization problem so far accepts or rejects edges based on marginal gain and constraint satisfaction in Step 5 of Algorithm 4. However, in practice, matching people to teams often also requires a lower bound of at least R^- people for each team. In Algorithm 4, it is possible that the cut-off is too high for marginal gain (Step 5) and enough people do not get assigned to each team. To solve this problem, we pre-calculate the marginal gains for each cluster and find the R^-th highest marginal gain among all clusters (denoted as df_{R^-}). This value is used to set the value of v (used in Algorithm 4) such that:

$$v \leq \frac{df_{R^-}.b.(1+2d)}{2}. \tag{3.10}$$

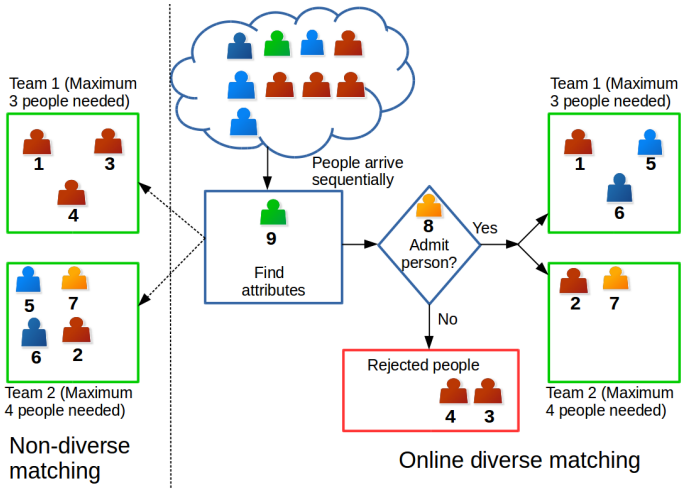

Figure 3.14: Diverse team formation workflow. People from different groups (represented by color) arrive sequentially (represented by numbers below them). The right side shows diverse team formation by our algorithm, while the left shows team formation by allotting people to teams in order of their arrival.

Setting v using Eq. 3.10 ensures that at least R^{-} workers will get accepted by the algorithm irrespective of the arrival order of people as the marginal gain of $(R^{-})^{th}$ person will still be below cut-off in Step 5 of the algorithm. In the simulation results, we explain how setting α or v not only helps ensure the lower bounds, but also improves overall matching utility. If the optimization problem in Eq. 3.9 is solved exactly with integral y, the current algorithm also provides a $\frac{\alpha}{1+2d}$ approximation of the optimal solution. The specific choice of v or order of arrival of nodes does not alter the theoretical guarantees provided in [281].

Figure 3.14 shows a toy example denoting the workflow of our algorithm in practice. Here, we need to allocate people to two teams (first team can accept maximum 3 people and second can accept maximum 4 people). People arrive in random order from a

large pool and belong to different groups unknown to us (shown by red, green, blue color). When a person arrives, we first find worker attributes (using interview or screening task) to help calculate the utility and diversity value they offer to existing teams. Next, they go to the decision making box, where Algorithm 4 is used to decide if they are rejected or allocated to one or more teams. Non-diverse matching is shown on the left side, which accepts people in the order of their arrival. One can notice that this leads to first team being formed of all people from same group, while diverse matching balances different groups for each team.

3.6.2.3 Performance Metrics for Diverse Allocation

We measure the performance of diverse matching on two factors — how much cluster diversity it adds to the task and how much utility it loses for the requester relative to maximum-weighted matching. To measure improvement in diversity, we measure the *entropy gain* (EG), as defined in Equation 3.3.

To measure the loss of utility due to diverse matching, we adopt the *price of diversity* metric proposed in Eq. 3.4 which measures the trade-off in economic efficiency under a diverse matching objective. Specifically, we define two complementary versions of this metric. First, to measure the economic loss due to rejection of people by diverse matching, we define the price of diversity ($\text{PoD}_{\#}$) as:

$$\text{PoD}_{\#} = \frac{\text{Number of people using diverse allocation}}{\text{Number of people using baseline allocation}}. \tag{3.11}$$

For example, let us say a team requires four people and diverse matching rejects

two people and finds an allocation after the arrival of the sixth person. If a baseline method accepts the first four people, $\text{PoD}_{\#}$ will be 1.5, implying that encouraging diversity requires interviewing/screening 1.5 times as many people. Normally, the cost of interviewing or screening candidates is low compared to the cost of the main task (*e.g.*, paying their salary); thus, even large values of $\text{PoD}_{\#}$ may be acceptable, and will also depend on resultant entropy gain. We also define utility-based price of diversity, PoD_u, to measure the aggregate weight lost due to rejecting people by diverse matching as:

$$\text{PoD}_u = \frac{\text{Utility obtained using baseline allocation}}{\text{Utility obtained using diverse allocation}}. \tag{3.12}$$

For example, say a task j requires three people, and that people belong to one of three clusters $k \in \{1, 2, 3\}$ with task utilities $w_{\{1,2,3\},j} = \{3, 1, 1\}$, respectively. If we use a greedy algorithm as a baseline, it will maximize utility only by selecting people from the first cluster, accruing total utility of 9, while diverse matching will accrue total utility of 5 by selecting one people from each group. Hence, PoD_u will be 1.8 against the greedy baseline.

3.6.3 Results and Discussion

In this section, we first test our proposed algorithm on simulated results, showing how price of diversity is affected by factors like utility and distribution of clusters. Next, we deploy it on an online platform to show how filtering works in practice. We collect data from 50 people for two online tasks. Using this data, we then show how our algorithm performs for different possible arrival orderings of workers for six different types of clus-

tering on demographic data.

3.6.3.1 Simulation Results

In this section, we consider simulated people sampled from different groups arriving online and assigned to different teams. We demonstrate the effectiveness of our method in different situations of varying cluster sizes, utilities and orderings.

We consider 10 teams, each of which requires at most 3 people. People are sampled from 3 clusters. While the total number of groups is known beforehand, a person's group or cluster id is known only after she arrives (*i.e.*, are interviewed/screened). The utility obtained from all people sampled from a group is the same. As people come from different distributions and groups have different weights, we first simulate a situation with equal weights for all clusters and equal probabilities. Next, we discuss clusters with different weights and how α affects matching performance. Finally, we show that our algorithm is robust, even for skewed distributions.

Clusters with equal utility: In this condition, we consider three equally probable clusters offering equal utility, where all people have unit utility for all tasks, hence $w = [1, 1, 1]$. We do 100 runs with a maximum of 100 people streaming in random order. People are drawn from a multinomial distribution with cluster probabilities $\theta = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$, respectively.

Solving the optimization problem in Eq. 3.9, we find $OPT^* = 30.0$. We set $\alpha = 1$, which gives the worst case performance bound of 1.428 for the online algorithm. Using Algorithm 4 to filter edges, we obtain the task assignment for all runs. In all 100 runs, we

were able to find a matching with utility 30.0, which is also the offline optimal allocation (one person from each cluster). Entropy for all tasks in all 100 runs is 1.09, implying that all teams were formed with people from three different clusters. We find that, on average over the different runs, the median number of people we need to interview before forming a diverse team is 5, with worst-case of 8 and a best-case of 3 people. Hence, median $\text{PoD}_{\#}$ is 2.67, while PoD_{u} is 1. This means that diverse matching improves coverage over clusters in all cases, but requires us to interview or screen 2.67 times as many people before we can form sufficient teams.

Avoiding Task Starvation: In our optimization problem, we do not impose lower bounds (cover constraints) on teams or tasks. However, for online team formation, teams may require at least R^{-} people to be effective. As discussed earlier, Eq. 3.10 can be used to set α, guaranteeing this condition. As $v_{min} = 31.5$ in the previous case, setting $\alpha = 1$ satisfied this condition.

α acts as a filter, as decreasing it lets the online algorithm accept more people from each cluster (forming less diverse teams for the sake of expediency) while increasing it accepts only the workers with highest marginal gain (holding out on candidates until it can form diverse team). On one hand, setting α too high will mean most people get rejected, leading to a matching where team never receive enough people. However, reducing α to very low values will essentially accept all people and behave similar to random team formation (*i.e.*, just allocate whichever person arrives first). For example, in the previous problem, when we reduce α to 0.4, the median fitness drops to 24.14, while the median entropy drops to 0.636. This means that the median team has three workers, who belong

to only two clusters.

Clusters with different utility: In this condition, we consider clusters with unequal cluster utility. This situation can arise when workers from a particular group are more useful for a given task than other groups. For example, unequal weights can be allocated to people when those from a particular group specialize in the task. In this work, we assume that we know the task utility for each group after the screening task and that other methods like expertise identification can be used to identify edge weights. In this simulation, we consider three clusters with utility $[3, 2, 1]$, respectively. We find that setting $\alpha = 1$ and simulating 100 runs leads to a median fitness of 31.4 with all tasks only matched to two people (one from cluster 0 and other from cluster 1). The optimal fitness from Eq. 3.9 is 42.42. However, if α is reduced to 0.7 (which is less than the cut-off of 0.74 calculated using Eq. 3.10), the desirable lower bound is met (each team receives three people) and the median fitness for 100 runs improves to 41.46 (which is also optimum fitness for the offline problem).

In this case, the median entropy is 1.09 with zero violations — *i.e.*, all teams get three people from three different clusters. On average, the team forms after 5 workers arrive. In the worst case, the team formed after 16 workers arrived, leading to a median PoD of 1.67. Fig. 3.15 shows how utility increases when lowering α initially, and then decreases on further reducing it. This is due to the submodular marginal gain of individual clusters as shown in Fig. 3.15. The x-axis shows the number of people selected from a single cluster for a single task. Here, each new person from a cluster provides less marginal utility and different clusters have different curves for marginal gain. Algorithm 4

accepts or rejects people if their marginal gain exceeds a cut-off directly proportional to α (as shown by the dotted red lines). We will accept people from a given cluster until the marginal gain curve for that cluster dips below the dotted line. For example, for $\alpha = 1$, only a maximum of one item from cluster 0 and 1 can be accepted, which always violates the lower bound by 1 unit for every task.

Similarly, for $\alpha = 0.3$, up to 5 people from cluster 0, 3 people from cluster 1 and 2 people from cluster 2 can be accepted. Although, the actual acceptance rate depends on order in which people arrive, setting α less than 0.74 guarantees that online diverse matching has zero violations as soon as one person from each cluster shows up. The theoretical lower bound on total utility in this case is ≤ 1.46 and in practice, we get much better results.

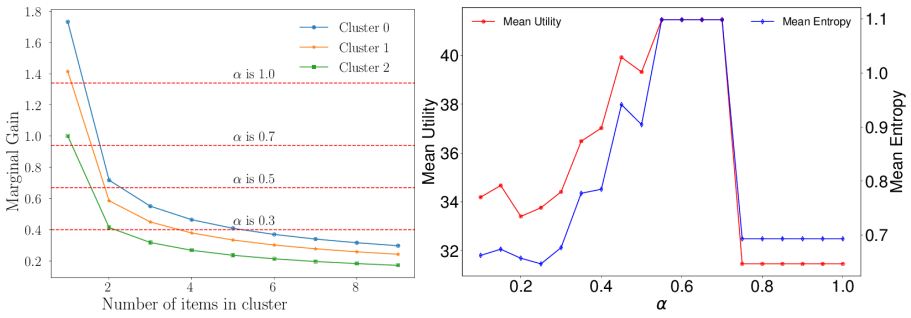

Figure 3.15: Left: Effect of α on worker acceptance from each cluster. Right: Effect of α on utility and entropy.

Different sized clusters: In real-world situations, people often have different probabilities of coming from different groups. For example, if a person wants to assemble a team of people belonging to different countries from an online community or pool for people (such as Mechanical Turk), and if we consider three clusters being the US, India and all

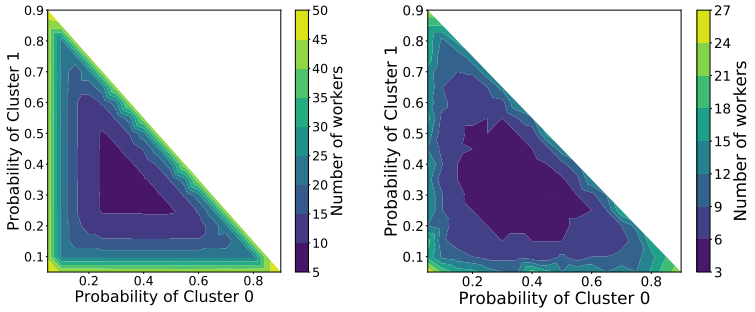

Figure 3.16: Left: Expected number of people needed. Right: Actual number of people needed (median of 100 runs).

of other countries, then past literature [82, 165] has shown that the distribution of these clusters on Mturk is $[75\%, 16\%, 9\%]$, respectively. This means that we can draw random people from a multinomial distribution with proportions $\theta = [0.75, 0.16, 0.09]$. Assume that the utility of assigning a person from these clusters to a team is $[1, 2, 3]$, respectively. If a firm knows these proportions, a natural and practical question to ask is "how much budget will I need to form a diverse team?" or alternatively, "how many people should I expect to reject in order to form a diverse team?"

To answer these, we use the following example. A firm can only pay to interview at most 10 people. When the firm starts interviewing, assume that $[6, 3, 1]$ people arrive from three clusters, respectively. As people are drawn from a multinomial distribution, we can calculate the probability of this event as: $Pr(6, 3, 1) = \frac{10!}{6!3!1!}(0.75)^6(0.16)^3(0.1)^{0.09} = 0.055$. We also know the maximum number of people allowed from each cluster (*e.g.*, one person), which means 7 people will be rejected in expectation. Likewise, we enumerate all possible scenarios for different numbers of people coming from each group, and calculate the expected number of people accepted for that distribution. In this case, we expect to

accept 2.95 people. This makes sense, as we need 3 out of 10 people to complete the task and in some cases people may arrive only from one or two clusters. As we increase the number of people we interview, the expected number of accepted people also increases. Hence, we can calculate the expected number of people we need to screen to get 3 people accepted for each team.

Figure 3.16 shows the expected number of people needed to get the desired three people (zero violations) for different cluster probability distributions. The x-axis shows cluster 0's probability while the y-axis shows cluster 1's probability. Even for very skewed distributions with $\theta = [0.9, 0.05, 0.05]$, we get a $\text{PoD}_{\#}$ of only 15.4. In context, if people are paid 10 cents to interview them compared to \$1 for actually doing the work, then for zero expected violations (*i.e.*, forming all teams) it costs only \$4.62 more compared to no screening and accepting the first three people — even under a highly skewed distribution of clusters with people from each of the two groups representing only 5% of the population. In the median case, where distributions are more even, it only costs an extra 50 cents to get diverse allocation. Fig. 3.16 shows the results on simulating 100 runs for different probabilities of clusters and observing the median number of people needed by our algorithm.

For clusters with different probabilities, we simulate 10 teams and 100 people, fix $\alpha = 0.7$, and calculate the utility and entropy for 100 different runs, drawing samples randomly according to the cluster 0 and cluster 1 probabilities. Each run randomizes the order in which people arrive. We find that, even for skewed distributions, our algorithm successfully finds high utility solutions. In all cases where people from all three clusters show up, diverse matching finds solutions as good as the offline optima. For edge cases,

ID	Age	Gender	Education	Country	Politics	Race
0	18-24 (20%)	Male (54%)	High school degree or equivalent (2%)	US (72%)	Democrat (46%)	White (56%)
1	25-34 (48%)	Female (46%)	Some college credit, no degree (12%)	India (28%)	Republican (30%)	Asian (30%)
2	35-44(14%)	-	Associates degree (12%)	-	Independent (20%)	Hispanic (2%)
3	45-54 (8%)	-	Bachelors degree (50%)	-	Other (4%)	American Indian or Alaska Native (6%)
4	55-64 (10%)	-	Masters degree (22%)	-	-	Other (6%)
5	-	-	Doctorate degree (2%)	-	-	-

Table 3.7: Distribution of various personal attributes in our MTurk experiment.

Cluster	Entropy Gain	$\text{PoD}_{\#}$	Median Entropy Gain	Median $\text{PoD}_{\#}$	Adversarial $\text{PoD}_{\#}$
Age	1.34	3.75	1.23	1.25	7.25
Gender	1.0	1.0	1.33	2	10.5
Education	1.33	2.0	1.33	2.25	9.5
Country	1.0	1.0	1.23	1.5	9.5
Politics	1.33	4.25	2.0	3.75	12.25
Race	2.0	10.75	2.0	3	11.25

Table 3.8: Algorithm's Price of Diversity ($\text{PoD}_{\#}$) and Entropy Gain performance in three cases: 1) Realized order, 2) Median case, and 3) Adversarial order.

where people from cluster 1, 2 or 3 never arrive, the competitive ratio (performance compared to the offline algorithm) is 0.81, 0.79 and 0.80 respectively. In these cases, our lower bounds are not satisfied by online matching as it only assigns two people per team rather than three. However, out of the $171,00$ total orderings we simulated, only 40 such violations occurred (*i.e.*, teams did not get three people as nobody from one cluster ever arrived).

As one would expect, the median number of samples needed for balanced distributions is low (5 people for $\theta = [0.33, 0.33, 0.34]$), while for skewed distributions, it is more (27 people for $\theta = [0.05, 0.05, 0.9]$). The values are similar to expected number of people shown in Fig. 3.16.

Adversarial (Worst-Case) Ordering: Suppose a firm is willing to interview 20 workers (some of whom they will hire), but it does not know how many people will come from each group. Assuming that the clusters have highly skewed utilities of $[1, 30, 30]$, the optimal worker allocation is $[0, 1, 2]$ people from first, second and third cluster respectively, with 13.2 utility ($OPT^* = 13.52$). We set $\alpha = 0.75$ for zero violations, which only accepts people from last two clusters due to the skewed weights. However, an adversary could have 20 workers from the lowest weight cluster (cluster 0 in this case) all apply first, and then our diverse matching strategy will not accept any applicants in this case. In contrast, if an unlimited stream of workers is allowed, we are guaranteed to have no violations and will achieve a utility of 13.2 when people from the second and third cluster eventually arrive. As we show next using real world data, even with an adversarial order, the price of diversity is not high in practice.

3.6.3.2 Crowd Evaluations

To test our algorithm in a real-world scenario, we implemented diverse worker allocation on MTurk via two stages. First, we posted a screening task where people provided us demographic information. For the sake of demonstration, we considered education diversity, under the assumption that we wish to form teams with different educational backgrounds. We show the categories and corresponding cluster id in Table 3.7. We categorized education up to a high school degree (ID 0) as Cluster 0, other non-graduate degrees (ID 1, 2, 3) as Cluster 1, and graduate degrees (ID 4, 5) as Cluster 2. This screening task filtered people using pre-set weights of $w = [3, 2, 1]$ and $\alpha = 0.7$. We designed a platform, which after receiving a person's screener response, either directs them to the finish page or allocates them to two different teams/tasks. Each team/task required 3 workers, we paid 10 cents for the screening task, and a $\$1$ bonus for the actual task. As an example, when we started the experiment, we received people with education levels denoted by the following labels (ID's in Table 3.7): $3, 1, 3, 1, 1, 4, 2, 3, 3, 3, 3, 4, 2, 3, 1, 3, 2, 0$. The first entry shows that the first person indicated her educational level to be "Bachelor's degree" (from Table 3.7), hence she belongs to Cluster 1, and so on for the remaining entries.

Upon running this experiment, we found that our algorithm accepted the first, second and eighteenth people, providing a diverse mix of education. Although the first three people could have provided a total utility of 6, they all belonged to the same cluster and offered no diversity of educational level (zero entropy as first three people had a similar education level). Our algorithm's diverse allocation also provided a utility of 6. However, it incurred a cost of $\$4.80$ rather than the $\$3$ it would have paid for non-diverse allocation.

$\text{PoD}_{\#}$ in this case is 6 and PoD_{u} is 1. Obviously, the actual price of diversity in different situations depends on the order in which people arrive.

To compare to counter-factual orderings, we ran another experiment where each person completed both tasks every time they accepted a job (*i.e.*, we did not perform online team formation). This allowed us to measure each person's performance on all tasks. We then used this data to evaluate our online algorithm by using a single data set to evaluate and compare several orderings/assignments. We provided people with two questions to gather community provided ideas as follows: 1) "How might we make low-income urban areas safer and more empowering for women and girls" 2) "How might we restore vibrancy in cities and regions facing economic decline?"

These questions were selected as they are open-ended, complex and accepts different viewpoints. They did not require previous knowledge. We ran the experiment in three batches (50 workers total). For the screening task, we requested demographics from each person regarding age, gender, education, country, political inclination, and race. In general, we observed that certain demographics were highly-skewed (*i.e.*, non-diverse) — see Table 3.7.

Table 3.8 lists the online matching results for three scenarios. First (column 2 and 3), we calculate the Entropy Gain and $\text{PoD}_{\#}$ for the actual order in which we received people. We consider each of the 6 ways to cluster individuals. The results show that we can achieve much higher entropy gain through diverse allocation. However, since the actual people we drew might not be representative of other possible orders, we took 1000 permutations of those people and re-calculated median values for $\text{PoD}_{\#}$ and entropy gain (column 4 and 5). On average, online matching successfully achieves large gains in

entropy. Finally, we calculate the adversarial ordering (worst-case scenario), where the smallest cluster shows up last. As expected, $\text{PoD}_{\#}$ is higher, but is not unreasonable due to the low cost of the screening task.

3.6.3.3 Discussion

Our above algorithms provide a scalable way to perform online, diverse, team formation that mirrors some of the constraints of real-world collaborative work and teams. However, our work leads to a number of open questions: 1) What kinds or types of diversity is our approach well- or ill-suited to include? 2) When in collaborative team formation would one want online diverse formation versus not? And 3) what kinds of diverse team formation tasks or constraints would limit the approach we outline here?

Handling Different Types of Diversity: Our above results demonstrated how to form diverse teams which were diverse with respect to people who were clustered into discrete groups (in our case, specifically based on demographics). However, our method is generic in the sense that it can be easily applied to any type of diversity wherein people can be categorized into a set of groups — whether it is based on demographics, task related skills, cognitive preferences, *etc.* There are two important cases that we do not explicitly handle above: 1) where people can belong to multiple groups/clusters (*i.e.*, where the clusters are not mutually exclusive) and 2) where there are not discrete clusters but rather continuous scales or spectra along which people vary.

When people may belong to multiple, non-mutually-exclusive clusters, one must modify our objective function in Eq. 3.9 to consider not just the given weight assigned to

that individual's group-to-team edges, but also other edges from other groups that the person may belong to. For instance, a person may have political affiliation as 50% Democrat and 50% Republican. If such a person gets matched to a team which tries to maximize diversity of political views, then all both groups get credit proportional to the percentage membership of the person. This increases the computational cost slightly (in that we have to consider more edges), but does not substantively change the above algorithm or results.

When people are mapped to a continuous or ordinal spectra (*e.g.*, right-to-left leaning, *etc.*) rather than in groups (*e.g.*, Democrat or Republican, *etc.*), diversity is often cast as a type of area, volume, or density coverage over a vector space. This changes the objective function — for example, using Determinantal Point Processes [162] instead of entropy over groups. In such cases, our greedy algorithm remains the same so long as the coverage function is submodular, but estimating OPT is more challenging. Methods for doing so are a fruitful area for future research.

Conditions for Diverse Team Selection: Theoretically, our proposed method applies to any situation where people belong to different groups and we want even coverage of those groups (*e.g.*, in team membership). However, practically, there are two important factors to consider. First is the price of encouraging diversity, especially in skewed distributions. In our simulated and human experiments, when some of the clusters or groups were quite rare, it was possible that requiring diverse matching rejected many people (while waiting for a person from a rare group to arrive). This rejection can have a non-trivial cost (*e.g.*, when interviewing people), which may affect the total budget. In such cases, one must balance the cost of rejection with the skewness of the applicant pool. If the cost of rejec-

tion is high or there are few applicants from a given cluster/group, then diverse matching can become expensive. In some situations, however, this higher cost may be worth the commensurate benefits of a diverse team.

Second, understanding that benefit-cost trade-off is central to knowing when and how to apply automated diverse team formation. Diversity is often portrayed as a "double-edged sword" in contemporary organizational theory [275]. At one end of the spectrum, proponents stress how heterogeneity helps team outcomes, while opponents posit that heterogeneous teams may lead to dysfunctional interactions or suboptimal performance. Different researchers in computer supported collaborative work community have studied diversity from the lens of creative output [238, 149], team satisfaction [278] or tie formation [84] *etc.* Although teams are routinely assembled from individuals with varying degrees of demographic and cognitive abilities, it is still an open question as to under what conditions heterogeneous composition leads to groups which outperform homogeneous teams [131]. While the answers to those questions lie beyond the scope of this work, our proposed method complements existing research on the benefits of diversity by allowing one to mathematically study whether balancing one type of diversity might be useful for a domain. For example, by calculating the "price of diversity," our method helps researchers in quantifying the impact of diversity on online team formation or other online matching problems.

As an example, consider two tasks. Task 1 requires a team to craft policies for an important national issue, while Task 2 requires the team to jointly write a review for the movie "Titanic." Assume that the manager wants to maximize diversity with respect to political affiliation (Democrats, Republicans, Independent, Others) for these two tasks.

As in our simulation studies, one can use population estimates to calculate the expected price of diversity. For instance, we observed a $\text{POD}_{\#}$ of 4.25 on Amazon Turk. This means, to form a team of 4 people for this task, we expect to reject another 13. Getting this estimate and comparing it to a firm's costs and internal values illuminates the pros or cons of political affiliation diversity for each task. For the first task, opinions from diverse political viewpoints will make the policy stronger and may be worth the rejection costs. On the other hand, current research does not indicate that political diversity substantially benefits dramatic movie review writing, and thus may not be worth the rejection cost. In such cases, the firm can decide whether more research is needed to establish the benefit or not.

3.6.3.4 Assumptions and Limitations for Research Task 3:

Below, we list the major assumptions and limitations of our online diverse team formation algorithm:

1. When a worker arrives, we assume that we know which cluster they belong to and after initial screening, we assume that we can estimate how much utility they will provide to the task. Estimating the utility of a person for a task is a difficult task, as the true utility can only be assessed after a person completes the task. This assumption limits the usage of our method to tasks where the value provided by a person cannot be estimated beforehand.

2. For every worker from a given class, we assume that they provide similar utility. This is a major limitation, as realistically different people from the same class may

provide different utility.

3. We assume that every worker who is allocated to the task will accept and complete the task. In reality, many workers drop out or may not complete the task. Future work can extend our algorithms to incorporate probabilities of task acceptance and task completion.

4. We assume that the submodular function in Eq. 3.8 is appropriate in capturing the diversity and quality of teams. However, other submodular functions may be more suitable for different types of tasks.

5. The goal of our online diverse matching algorithm is to maximize the total utility of all tasks. This global optimization assumes that every task or team accepts or rejects people to maximize global objective. However, each team may maximize its own objectives (trying to get the most suitable workers or the most diverse team for itself) leading to globally suboptimal solution. This presents a limitation of our work, where a centralized decision-making body is needed to allocate workers to teams to maximize the total objective.

From the simulations provided, one may wonder why a computational method is needed at all. Can diverse matching just be done manually? For a small number of tasks and clusters, where all team members are equally qualified for the tasks, it is possible to form diverse teams manually. However, when the constraints are more complex (*e.g.*, different tasks have different demands, multiple clusters exist, and different people have different utility) it quickly becomes impossible for a human to select diverse teams. In such cases, our diverse team formation method applies.

Another important implication of our research lies in a better understanding of team member utility. In our simulations, we assumed that we already knew the edge weights or the utility that a person offers to all the tasks. In practice, it is non-trivial to estimate that utility and a large body of research have looked into estimating a person's task utility [21]. Future research directions can look at this problem holistically, to estimate utility for diverse teams. One interesting direction would be integrating online diverse team formation with simultaneous utility assessment (*e.g.*, based on worker accuracy in crowd markets).

Likewise, one must estimate a person's cluster or group. This work used demographic groups but our method allows groups based on any factor. With some modification to the objective function, it is possible to allow multiple group membership too. However, defining groups in itself are non-trivial for some applications, and a person's group, affiliations, or characteristics may change over time. These questions complement our line of work and would be interesting areas for future research.

Extensions beyond Team Formation: Thus far, we have discussed how to form diverse, collaborative teams. However, team formation can benefit from diversity in two different ways — by joint team effort or just by aggregating individual efforts. For the former, organizational research has investigated many factors where diversity may benefit team output. However, a less obvious application of diverse team formation is the scenario where the team members work independently. In such cases, one expects to benefit from aggregating their individual outputs to form a collective output. Conference or journal paper reviewing is one example of this situation, where reviewers are not necessarily collaborating together, but aggregating reviews from diverse viewpoints will benefit a

paper more than those from the same viewpoint. Diverse matching also applies to such broader definitions of team tasks. For instance, many online design communities expect participants to also review and critique each others' designs [100, 98]. By matching diverse sets of individuals to each design, one can expect to get reviews from different viewpoints. Online matching is necessary in this case as people arrive randomly over time and need a subset of designs to review. Similar issues arise in network science and formation as well, such as the preferential attachment problem.

3.6.4 Concluding Remarks of Research Task 3

We presented a method for assigning people from different groups to teams — online diverse matching. We show that by using a low-cost screening task, one can group people and then allocate them to teams as they arrive while balancing the team diversity. While we clustered people into groups based on demographics, our method is generic and can be applied to other attributes like expertise. Our method also applies to other online allocation tasks where diversity of viewpoints might matter: *e.g.*, online worker-to-task assignments, journal paper-reviewer assignments, and intelligence analysis tasks. Future work could include: 1) journal paper-review assignments where both the static and dynamic side of the bipartite graph are clustered; 2) latent or non-mutually exclusive cluster labels/attributes; and 3) combining online diverse matching with online cluster identification using Bayesian techniques [191].

3.7 Key Contributions

This work studies the trade-off between diversity and efficiency in matching markets. This is different from earlier work as the diversity measure is modeled as an objective and not as constraints, and diversity is defined over *sets* of items. The main contributions of our work from three research tasks are as follows:

1. By using a quadratic function to measure coverage, we formulate the diverse weighted bipartite b-matching optimization problem which can be solved using a Mixed Integer Quadratic Program.

2. We provide the first anytime PTIME algorithm for the diverse bipartite b-matching problem with class-specific weights.[3] The key insight lies in detecting *negative cycles* in the matching graph, which we use to either provide incremental improvements to the incumbent diverse matching, or prove that our negative-cycle-detection algorithms have found a globally-optimal matching.

3. We show via simulation and data from three large real-world bipartite matching problems that our method produces matchings with much higher diversity than standard efficient matchings, at a little overall cost to economic efficiency. These findings demonstrate that the benefit of incorporating diversity outweigh the cost in many applications.

[3]That is, under conditions when the utility of assigning all items from one category to an item on the other side of the graph is the same. This holds when, e.g., one is matching academic papers to reviewers where each reviewer can specify exactly one field of expertise and the utility of assigning a paper to any of the reviewers *within* the same field is the same, but differs *across* fields.

4. We show how concave submodular functions can be used to formulate the online diverse bipartite b-matching problem as an optimization problem. We demonstrate how our general formulation resolves, as a special case, to online worker-to-team matching.

5. We overcome issues (like estimating optimal objective value apriori) with existing online submodular optimization methods to present a simple approximation algorithm for performing online diverse matching.

3.8 Directions for Future Work

In this chapter, we laid the groundwork of diverse team formation by proposing computational approaches to forming teams, both when workers arrive sequentially and when a group of workers are present in a pool. This body of work has thrown light on many new research directions, some of which are highlighted below for offline and online matching.

- Learn submodular function which will reflect how users perceive diversity.
- Improve the efficiency of offline diverse b-matching by proposing a new greedy algorithm. We are looking into the option of using edge swaps across clusters to locally improve diversity.
- Find the conditions in which diverse teams are useful or to answer how much diversity is needed for any given domain?
- Define the diversity of a set either using functions learned from humans (to better model diversity) or using Determinantal Point Process (DPP) kernels [162], which

do not require explicit clustering.

- Another direction of research can be studying the trade-off between diversity and efficiency for different applications — to estimate the trade-off in combining efficiency maximizing WBM and entropy maximizing D-WBM.

- Many real-world applications have uncertainty. The classical results we used in our work consider the oracle model whereby the access to the submodular optimization objective is provided through a deterministic function. However, in many applications, the objective has to be estimated from data and is subject to stochastic fluctuations. For our matching application, the edge weight estimation may have fluctuations, which requires solving stochastic submodular maximization problem. Stochastic gradient methods [142] have recently been suggested to solve problems with cardinality constraints. Future work can investigate how we can solve diverse matching problem using similar methods.

- The online diverse matching currently approximates an upper bound of the optimal solution. We plan to use recently proposed Mixed Integer Convex methods to improve the approximation.

- Combining online diverse matching with online cluster identification using Bayesian techniques [191].

- Future work could explore the extension of this method to online diverse matching [81], where vertices arrive sequentially and must be match immediately; this has direct application in advertising, where one could balance notions of reach,

frequency, and immediate monetary return. Exploring connections to fairness in machine learning [110] and hiring [231] by way of diversity are also of immediate interest.

However, these are broad questions, which may encompass several Ph.D. theses. In this dissertation, we work under the assumption that it is known (say from past experience) if diverse teams are better or not for the task at hand. Nevertheless, answering these questions will help users in practical applications of our algorithms.

3.9 Conclusion of Chapter 3

In this work, we presented quantitative approaches to balancing diversity and efficiency in a generalization of bipartite matching where agents on one side of the market can be matched to sets of agents or items on the other. To solve the offline problem, we propose a quadratic programming-based approach to solving a supermodular minimization problem that balances diversity and total weight of the solution. The general problem is NP-hard, so we proposed a scalable greedy algorithm with theoretical performance bounds. We proposed the price of diversity (PoD), which measures efficiency loss due to enforcing diversity, and gave worst-case theoretical bounds on that metric. Finally, we validated our methods on three real-world datasets and showed that the price of diversity is quite good in practice. To further improve the MIQP approach, we proposed the first pseudo-polynomial-time algorithm for diverse weighted bipartite b-matching. We showed that our algorithms not only guarantee optimal solutions, but also converges faster than the existing state-of-the-art approach using a black-box industrial MIQP solver.

To solve the online problem, we presented a method for assigning people from different groups to teams. We show that by using a low-cost screening task, one can group people and then allocate them to teams as they arrive while balancing the team diversity. While we clustered people into groups based on demographics, our method is generic and can be applied to other attributes like expertise. Our method also applies to other online allocation tasks where diversity of viewpoints might matter: *e.g.*, online worker-to-task assignments, journal paper-reviewer assignments, and intelligence analysis tasks.

In this chapter, we proposed algorithms to form diverse teams. Our goal was to allocate reviewers to ideas, such that each idea receives reviewers who are different from each other and suitable for the reviewing task. These algorithms can be used by organizers of an online design contest to get review scores for each idea. The next goal is to select a small set of winning ideas which are funded or implemented by the organization. In the next chapter, we propose optimization methods to filter design ideas. We first investigate how diversity needs to be measured, learned and optimized for ranking of a set of items. We also introduce Determinantal Point Processes [162], which provide a way to measure diversity in continuous space, when discrete cluster labels may not be available. Finally, using simulations and crowd experiments, we show these algorithms are better than existing methods in filtering ideas.

Chapter 4: Diverse Idea Filtering: How does one filter good ideas out of hundreds of submissions?

In this chapter, we explore the question, "How does one filter good ideas out of hundreds of submissions?" Good ideas are often defined as high quality, novel and diverse. To answer this question, we show how to rank order ideas for diversity and quality by formulating it as an optimization problem and then using multi-objective optimization methods to solve it. In the second part, we demonstrate how diverse ranking improves efficiency in real-world idea filtering tasks by deploying the system for two crowd evaluation tasks. Overall this chapter formulates mathematical methods to measure diversity of a ranked list of items and provides ways to optimize that list. It also demonstrates practical applicability of diversity metric in idea filtering tasks.

We divide our work into two research tasks:

- Research Task 1: Ranking ideas for diversity and quality
- Research Task 2: Filtering innovative ideas using diverse ranking

In the first research task, we discuss algorithms for diverse ranking of a set of items. In the second research task, we show how a diverse ranking algorithm can be combined with a novel crowd voting mechanism for efficient filtering. In the next section, we discuss the background and motivation for our research on idea filtering. Thereafter, we discuss

how other people have approached parts of this problem. Finally, we show our results and discuss possible future extensions.

4.1 Background and Motivation

Open innovation is the process where an organization opens up to external parties (customers, stakeholders, volunteers) to gather out-of-the-box ideas on how to solve challenging problems. The rationale is that while many problems can be solved within the traditional boundaries of the firm, sometimes the knowledge, intuition and radical creativity required for solving new problems is not available and must be sought outside. In the last decade, open innovation has led to a major shift in how we think about R&D: from siloed, in-house discovery to the engagement of external crowds, with leading firm s (like Intel, Cisco [123] and Lego [184]), semi-autonomous organizational collectives [111], and innovation brokers (OpenIDEO, Innocentive[1]) all relying on it to select the projects to be funded.

However, one of the main – and persistent – problems that organizations face after an open innovation idea contest is that of filtering ideas. Contributors have just sent in a flood of candidate solutions of variable quality, and these solutions must now be reviewed, and the most promising among them must be selected. At this stage, organizations usually rely on in-house experts who will evaluate and filter the ideas. This can be a cumbersome, costly and lengthy process, which creates significant production bottlenecks and increases the transaction costs for searching and evaluating externally-sourced knowledge [148]. An indicative example is Google's 10 to the 100^{th} project, which received

[1] https://openideo.com/, https://www.innocentive.com/

over $150,000$ suggestions on how to channel Google's charitable contributions [262]. To deal with the unexpected deluge of submitted ideas, Google had to allocate 3000 employees for the filtering of the ideas; a process that put them nine months behind schedule. Furthermore, research has shown that the in-house experts may miss good solutions due to the significant cognitive load involved in reviewing multiple diverse ideas in a short time frame [265, 250].

The problem of a large collection of ideas is not only faced by organizations, but designers face the same problem during ideation contests. When generating creative designs, both practicing designers and researchers agree: "If you want to have good ideas, you must have many ideas." [206] Why? Because having many ideas helps a designer — or a team of designers — explore design space and find new inspiration from unlikely places. But is more always better? When does 'many ideas' turn into 'too many ideas'? Together, these problems cause serious concerns about the practical usability of open innovation and *often make organizations, as well as investors reluctant about using open innovation altogether* [169, 133].

Given thousands of possible ideas to process and limited time, a designer needs a much smaller "good" set of seed ideas or, better yet, a good ranking of all ideas so that they can decide when they have had enough. But what, specifically, does it mean for a ranking of ideas to be "good", how does one compute such rankings? and how do such rankings help in improving open innovation? In this research, we focus on those three questions. Specifically, we argue that when ranking ideas — *e.g.*, for the purpose of inspiration, idea generation, or filtering — a good ranking should not only show a designer ideas that possess high *quality* — that is, ideas that perform better than other

ideas (assuming one can measure such differences accurately) — but also that possess *diversity* — that is, a designer should see ideas that *cover* a design space well.

Why would one care about encouraging diversity when ranking ideas? Why not just order ideas by individual quality or merit and be done with it? Consider the following example design problem from a real-world design competition[2] which asked designers to generate ideas to address "How might we better connect food production and consumption?" Of the 606 submitted ideas, let us take a summary of just four ideas as an example:

1. Compost It! — A proposal to partner with the city to create a closed loop composting system.[3]

2. Residential compost material — curbside pickup — A state-wide initiative to encourage people to separate compost material for pick up.[4]

3. The Art of Food Festival — A festival celebrating local food and art with edible sculptures, inspired by french festivals. [5]

4. Online local farming NFP organisation — Growing and delivering fresh locally grown vegetables to a community of online customers at a very low cost. [6]

The above ideas have quality scores — provided by human raters — of 20, 12, 9, and 3 points respectively. Let us say that our task is to show two "good" ideas to a designer

[2] `http://challenges.openideo.com/challenge/localfood/`

[3] challenges.openideo.com/challenge/localfood/concepting/compost-it

[4] challenges.openideo.com/challenge/localfood/concepting/residential-compost-material-curbside-pickup

[5] challenges.openideo.com/challenge/localfood/concepting/the-art-of-food-festival

[6] challenges.openideo.com/challenge/localfood/concepting/online-local-farming-nfp-organisation

where a "good" set of ideas should help inspire the designer to come up with new ideas. One naive way is simply to order all ideas by their quality score and select the top two ideas. However, in our example, this will select the two ideas related to composting. Is this a good choice?

On the one hand, they are the two highest quality ideas of the four.[7] On the other hand, they are surprisingly similar to each other; both address the fairly broad problem statement — connecting food production and consumption — via a narrow set of solutions — composting. As many researchers have shown, generating good ideas requires both divergent and convergent thinking, and it is not clear that ranking purely by quality promotes such divergence. Likewise, if quality ratings are biased or noisy, promoting coverage may protect against unfairly discounting certain ideas. Ideally, selected ideas should have *both* high quality and good coverage of possible options. This allows a designer to gain maximal benefit from a large number of ideas — *e.g.*, increased coverage and quality — within a given budget of time or attention.

How does one find high-quality ideas that also have good coverage? One manual approach might first rank ideas by their quality and then just swap ideas which are similar to each other with random ideas from the collection. For our above example, the first two ideas are similar, so we can swap the second idea with either the third or fourth to get a diverse set of two ideas. But when the number of ideas grows to hundreds or thousands this approach does not scale; finding exactly which ideas to swap in is laborious and depends on the other ideas you already have in the set. Astute readers may notice that, mathe-

[7] Assuming (perhaps tenuously) that our measurement system, be it, humans, computational simulations, analytical formulas, *etc.* is not noisy, biased, or fixated towards particular solutions like composting.

matically, this is equivalent to a combinatorial optimization problem called *set covering* which is a type of boolean satisfiability problem. Optimizing such problems is NP-Hard. The second approach, and one which is commonly used, is to define an objective function which is a weighed average of diversity and quality. While this approach is straightforward to implement, it is difficult to know beforehand how much quality one is willing to part with to encourage diversity. Finally, the approach we use in our first research task formulates a multi-objective optimization problem and treats coverage and quality as independent objectives. One benefit of doing so is that after computing the trade-off front one can actually compute the loss in quality for any given gain in coverage.

In addition, as different designers may have different information needs, instead of selecting a smaller subset of two ideas and showing them to a designer, one can also *rank order* all ideas. This retains all ideas where the ones appearing on top of the list are good (*i.e.*, higher quality with good coverage). Deciding what ranking is better is non-trivial. Even for our simple example, it is hard to argue which of the following rankings is clearly better: [1,3,4,2] or [1,4,3,2] or [1,3,2,4]. While, at first glance, ranking ideas may seem straightforward, including diversity transforms ranking into an NP-Hard problem.

In addition to meeting the information needs of designers, researchers have recently started to explore using the crowd to filter the candidate ideas for organizations too. The most typical strategy used by popular open innovation platforms like OpenIDEO is majority voting, where a user can go through the candidate ideas and upvote their preferred ones, and ranking is dynamically[8] calculated in a descending vote order. The problem

[8]Dynamic ranking means that the number of votes per idea is updated every time a user casts a vote, and this information is used to re-calculate the ranking that a new user sees when he/she first enters the

with this strategy is that it is prone to quickly locking into a fairly static and arbitrary ranking of the ideas because of positive feedback loops, as people tend to fixate on the few ideas that have already received good ratings or are readily visible [225]. For example, in the OpenIDEO challenge with which we will work on in this chapter, half of the crowd's evaluations went to only 10% of the ideas, while one fifth of the ideas (21%) received no evaluation[9]. A second problem with majority voting is that the crowds are less effective in distinguishing mediocre from excellent ideas [154].

To address the issues of ranking and filtering, in this chapter we propose a) a diverse ranking method which outperforms existing benchmarks and b) propose DBLemons: a crowd-based idea filtering strategy which helps increase filtering efficiency by balancing idea quality and idea concept space coverage. We compare DBLemons against two ranking strategies: i) majority voting, which replicates the standard voting mechanism used in today's online innovation communities and ii) Bag of Lemons [154], a state-of-the-art approach that uses negative instead of positive voting, which we extend for use in the dynamic ranking setting of real-world innovation platforms.

4.2 Literature Review

4.2.1 Ranking Ideas

Two seemingly disparate fields — Design and Computer Science — have both explored ways to jointly rank quality and diversity. Design researchers have focused on appropriate

platform. We will use this definition throughout the chapter.

[9]OpenIDEO challenge on women's safety: `https://challenges.openideo.com/challenge/womens-safety/refinement`, Evaluation stage

metrics for measuring item diversity and quality, while Computer Science researchers have focused on representations and methods for scalably estimating or ranking lists of diverse items. This work advances different efforts across both fields.

Within Design, researchers have primarily tackled how to either (1) evaluate creative sets of ideas or (2) leverage large design databases to inspire designers. As an exemplar of the former, Shah *et al.* [234] provide metrics for ideation effectiveness, where the main measures for the goodness of a design method are how they expand the design space and how well they explore it. Typically, work in this vein discusses diverse design space exploration using terms like *variety*, measured through, for example, coverage over trees of functions [234, 268], human expert assessment [122], or linear combinations of design attributes [99]. One of the differences between past engineering design variety literature and what we propose is that many past variety measures require expert coding for all ideas, which may be infeasible for a large collection.

The second main avenue of research concerns evaluating large sets of ideas, typically by using crowds of evaluators to scale up evaluation by partitioning ideas among many people. As an exemplar of such approaches, Kudrowitz and Wallace [159] suggest metrics to narrow down a large collection of product ideas. Likewise, Green *et al.* [108] propose methods for creativity evaluation using crowd-sourcing, where researchers focused on inspiring designers [270] and inspiring creativity [64].

Within Computer Science, researchers have tackled diversification in two strongly inter-connected applications: information retrieval and recommender systems, where researchers have developed ranking algorithms for different settings. When recommending sets of items to people (*e.g.*, movies on Netflix) predicting exactly what a user wants is

difficult, so by recommending a diverse set of items, chances increase that one of the recommended items will match what the user wants. The intuition for this approach stems from the *portfolio effect* [20] where placing similar items together within a portfolio of items has decreasing additional value for users. This *diminishing marginal utility* property is well-studied in consumer choice theory and related fields [68].

The main research questions within both recommender systems and information retrieval are two-fold: (1) how do we represent this diminishing marginal utility, and once we do (2) how do we optimize over it efficiently? For the former question, researchers have proposed alternate scoring methods to diversify rankings. An early exemplar of this was Ziegler *et al.* [290] who modeled the topics in text documents and then tried to balance the topics within recommended lists. Their large scale user survey showed that a user's overall satisfaction with lists depended on both accuracy and the perceived diversity of list items. Approaches that followed largely centered around the notion of *coverage* — that a diverse set should somehow cover a space of items well. The main differentiators of past approaches are how this coverage is measured and then combined with other objectives such as document relevance.

Approaches to measuring coverage break into two main camps, depending on what objects the coverage is defined over. The most common approach defines a vector space using properties of each item, *e.g.*, word frequency vectors or topic distributions over text. For example, Puthiya *et al.* [214] take positively rated items from a user, and then select sets from that list such that they cover the distribution of words in the submission. Likewise, search diversification techniques such as xQuAD [227] explicitly model the underlying aspects or subtopics for a query and select documents based on a combination

of their relevance to the original query and relevance to the aspects.

The second camp instead defines a similarity graph between items — for example cosine similarity between documents — and then computes properties over this graph such that the selected items maximize some graph coverage property. For example, one can use random-walk based algorithms like PageRank [285, 120] to estimate how central items are in a graph, and then re-order items based on this score. For more examples of such variants, Vargas *et al.* [266] and Castells [52] provide useful frameworks and reviews of past approaches. Such approaches apply to a broad set of applications like music discovery [287], keyword-based summarization [95], ecology[205], and document summarization [289].

Assuming we can answer the former question — how to represent diminishing marginal utility of sets — the latter question concerns computing such rankings. Three difficult and inter-related problems have motivated past research: (1) there are different ways of computing *coverage* over a space — under what conditions would we prefer one over the other? (2) Coverage over *sets* of items is a combinatorial problem (optimizing set-cover is NP-Hard) — how can we guarantee certain performance in polynomial time? And (3) diverse rankings require some notion of optimal coverage across a ranking, which is harder than guaranteeing coverage over a single fixed-size set — how should we compare optimal coverage over such rankings?

For the first problem of which coverage metric to use, researchers have proposed many different options. However not much work has characterized and compared the differences between these options; this is one of our work's contributions. For the second problem, most work has focused on using greedy approximations to the set coverage

problem. This means most of these methods produce a list by progressively adding items to a set, with some fixed weighted trade-off between diversity and relevance [288]. While this efficiently produces diverse lists, it is difficult to compare or customize such lists when users have different preferences between diversity and quality. One of this work's contributions is to provide, to our knowledge, the first approach to compare entire ranked lists between these two objectives and efficiently create rank orders that span the trade-off between diversity and quality (Sec. 4.4.2). For the third problem, past work typically considers rankings more diverse if they minimize some notion of redundancy. For example, whether ranked items occur in common elements in a hierarchy [272], or how well rankings compare with human relevance judgments of sub-topics such as ERR-IA [57], α-nDCG [66], and S-precision or S-recall [50]. These metrics are difficult to extend to cases where we do not have human-provided labels. One of this work's contributions is to extend coverage metrics used for fixed-size sets to rankings, such that we can use those metrics to evaluate the diversity of ranked lists (Secs. 4.4.2.1 and 4.4.2.2).

Compared to information retrieval or recommender systems, where the number of sub-topics is frequently set in advance and users have a specific query they wish to answer, design ideas are often unstructured, come from a wide variety of sources, and a designer's goal is to gain inspiration from a wide range of sources. This makes generating diverse, high quality lists particularly important when providing ranked ideas to designers. If successful, such techniques would have wide ranging consequences for crowd-sourced or large-scale ideation techniques by helping designers avoid premature convergence on a very limited set of ideas and helping people explore vast design spaces.

4.2.2 Idea Filtering

4.2.2.1 Idea Evaluation: Who will be the reviewer?

Machines or Humans. Ideas can be judged using machines, humans or a combination of the two. Machine rating is generally applied to ideas that are in a structured format. For example, ETS grading [87] uses natural language processing to grade papers. However, it is still difficult for machines to evaluate ideas on aspects like creativity, as this requires combining high-level knowledge from heterogeneous sources. In contrast, human intelligence excels at acquiring, understanding and making mental connections among diverse knowledge sets, and in making abstractions. Although very recent literature has moved towards the direction of teaching machines how to perform these tasks [129], *humans are still the best option for intellectual, subjective tasks like innovation assessment.*

Experts or crowds. Human evaluators can be experts or non-experts. Experts have substantial knowledge of the field and of the market, and can thus provide more informed and trustworthy evaluations [62]. Many crowdsourcing platforms such as Topcoder, Taskcn, and Wooshii use expert panels to select contest winners [61]. However, experts are also scarce and expensive, since gaining expertise on a particular innovation subfield takes a substantial amount of training. In practice, convening an expert group to evaluate a large number of ideas of an open innovation contest has proven to be prohibitively slow, costly, and to cause significant decision and production bottlenecks [153]. Crowds have been proposed as an alternative to evaluating ideas that require human input. Apart from the evident advantage of being faster and more cost-effective [176], adequately large numbers

of people have proven to make accurate estimations of reality due to their large diversity of viewpoints, knowledge and skills ("wisdom of the crowds" notion [249]). Expertise can also be found within the crowd, and researchers have searched for ways of leveraging it through expertise-weighted consensus mechanisms [46], and priming techniques for novices to serve as expert proxies [203]. Under the right conditions, crowd-based idea evaluation has been shown to be equivalent to that of experts [151]. Evidence finally exists that crowds can provide high quality opinions on difficult judgment and choice tasks, frequently outperforming individual experts [91]. *Overall, recent literature shows that crowds have potential in evaluating promising ideas. Combined with the prospect for reduced cost and faster turnaround times, they constitute an alternative that needs to be explored for high-quality idea filtering.*

4.2.2.2 Idea Evaluation: How will the ideas be selected?

Author-based or content-based. Idea evaluation mechanisms can be broadly classified into author-based or content-based [152]. In author-based evaluation, the reputation of the author is the main determinant of selecting good ideas. This reputation may be known a-priori or it may be gradually assessed over longitudinal tasks [44]. Ideas from reputable authors or authors scoring high on standardized questions are selected. Content-based filtering places focus on the idea itself and its potential to effectively solve the given problem. This mechanism can be further divided into two evaluation mechanisms: i) judgment-based or ii) choice-based.

Judgment-based or choice-based. Judgment-based evaluation involves individually assessing each idea on a pre-agreed rating scale (e.g. Likert) [219, 75]. It offers the advantage of a meaningful interpretation of the result, since all ideas are assessed against an absolute standard (the scale's endpoints). It has also been connected to higher perceived ease of use and higher decision quality [37]. However, because each idea is assessed individually, judgment-based evaluation takes time and can thus be expensive [153]. *In this work we will use the judgment-based approach to create the benchmark dataset of ideas, which we will use to compare the three crowd-based filtering strategies that this work investigates.*

Choice-based filtering involves comparing a set of ideas and then, in accordance to certain evaluation guidelines, selecting some of these ideas based on one's own preference [190, 208], or based on the expected preferences of other evaluators [37]. Because it involves comparisons and not individual assessment, choice-based filtering is faster and thus more cost-effective for use by large numbers of evaluators [271]. However, since it does not evaluate ideas against an absolute criterion, its performance is affected by the ranking strategy, i.e. the order in which the ideas are presented to the crowd. *In this work we propose a new choice-based filtering strategy and will compare it with two real-world and state-of-the art alternatives.*

4.2.2.3 Idea Filtering Mechanism

Majority voting In majority voting, each voter gets to upvote ideas they like (similar to Facebook "Likes"). Ideas are ranked in descending order based on the number of upvotes they receive. This method enables a crowd to provide similar accuracy to a Likert scale-

based evaluation but at a fraction of the required time [153]. However, it also presents two problems when applied to large idea collections. First, the "snowball" effect, where voters are fixated on a few ideas or idea concepts (ideas with similar thematic) because these ideas or concepts received the first initial upvotes and thus are more likely to be seen (and voted on) by subsequent users, while other potentially better ideas do not receive an equal share of attention [225]. Second, when positive voting is used, as in majority voting, users are less likely to distinguish mediocre from excellent ideas [154]. Despite the above, majority voting is a straightforward method for voters to understand, and thus it is widely used by many online open innovation communities like OpenIDEO and GrabCAD [225].

In Research Task 2, we will replicate majority voting and use it as a baseline for comparison with the other two ranking strategies with which we will experiment.

Bag of Lemons To address the second problem of majority voting, i.e. the difficulty of crowd voters to filter mediocre from excellent ideas, Klein and Garcia ([154]) introduced the notion of "Bag of Lemons" (BoL). The key insight is that crowds are better at eliminating bad ideas than they are at identifying good ones. In their experiment, a group of lab members was informed that a given set of ideas had been reviewed by an expert committee, and that their job was to predict which ideas had been selected as winners. They used the BoL multi-voting technique with static ranking (ideas are not re-ordered as participants vote on the platform) and compared these results with the Bag of Stars and Likert scale approaches. They found that using the Bag of Lemons approach provided a better recall/compression trade-off with significant time improvements. Other literature on BoL [271] has looked into user activity when using BoL and compared it to both the

Likert scale and up/down-voting.

In this work, we extend the BoL approach on a *dynamic voting setting*, where multiple voters arrive at different times, with unknown arrival rates, and where each voter views the ideas ranked according to the votes of the previous users. In this setting, which is closer to the actual conditions of real-world open innovation communities, we explore whether Bag of Lemons will still manage to increase filtering efficiency and reduce task time for the crowd workers.

4.2.2.4 Cross-Domain Inspiration: Idea Diversification

To address the first problem of majority voting, i.e. that voters tend to be fixated on a few idea concepts while others receive disproportionately less attention, we draw inspiration from the notion of *diversity* discussed before. Diversity is most often encountered in the fields of information retrieval and recommender systems, where researchers seek to recommend interesting sets of items to people (*e.g.*, movies on Netflix), and where predicting exactly what the user wants is difficult. One strategy around this is to recommend a diverse set of items, hoping that by covering a diverse space of options, the chances of matching one of the recommended items to user preferences will increase. The intuition for this approach stems from the *portfolio effect* [20] where placing similar items together has decreasing additional value for users. This *diminishing marginal utility* property is also well-studied within consumer choice theory and related fields [68]. Various approaches have been proposed for representing and optimizing this diminishing marginal utility to achieve efficient diversification [266]. Such approaches relate to a broad set of applications like music discovery [287], keyword-based summarization [95],

ecology[205], and document summarization [289]. In this work, we examine if diversifying the idea ranking based on thematic concept clusters (sets of ideas with similar thematic) and combining this with the BoL strategy (which has proven to be better than majority voting at least on static settings) will ensure a better coverage of the idea space and thus help increase filtering efficiency.

4.3 Research Gaps and Research Objectives

In building on related literature, we found a few gaps. Firstly, submodular functions [171] and Determinantal Point Processes [162] have been proposed in the literature to measure the diversity of a set. However, we did not find principled ways of using them for ranking a set of items. The ranking of ideas is a harder problem than measuring diversity of sets, as it requires comparing the diversity of sets of different sizes. Secondly, we found that Bag of Lemons (BoL) multi-voting strategy has been shown in literature as a promising way to filter ideas. However, the method has not been evaluated on real-world ideation contests, which typically have ideas ranked dynamically. Finally, diversity has been shown in the literature to help designers in divergent thinking, but the application of diverse ranking for idea filtering has not been explored yet. To address these gaps, we try to answer the following questions in our work:

- How to compute balanced high-quality diverse ranking in discrete and continuous space?
- Does Bag of Lemons (BoL) outperform majority voting in filtering efficiency within a dynamic vote ranking setting?

- Does diversity-assisted ranking increase BoL's efficiency in filtering high-quality ideas?

4.4 Research Task 1: Ranking Ideas for Diversity and Quality

In this research task, we investigate the question, "How to compute balanced high-quality diverse ranking in discrete and continuous space?"

4.4.1 Defining and Computing Diversity for Fixed-Size Sets

Before we can address *ranking* ideas by diversity, we first need to introduce how to quantitatively compute the diversity for simpler *fixed-sized* sets of ideas. For example, when one needs to pick a diverse set of five ideas, but the exact order in which one picks them does not matter.

Consider the example from the beginning of the chapter, where one needs to select two ideas out of four related to "connecting food production to consumption." In that example, one can intuitively tell that selecting the first two ideas — both relating to composting strategies — seems less diverse than the first and third ideas — one on composting, and one on food festivals. Why does one conclude this? How can we make this intuition more precise? Can we quantitatively capture that intuition?

As with the related work summarized above, quantitatively measuring diversity essentially comes down to measuring how well a set of ideas *covers* a *space of options*. For our above example, one might look at the four ideas and mentally place them into "buckets," placing the two composting ideas into the "compost" bucket, the food festival

idea into an "events" bucket, and the online farming group into an "online community" bucket. Computing diversity — or how well a set covers a space of options — might then translate into calculating whether selected ideas come from different buckets.

Alternatively, one could imagine printing out the ideas, placing them on a table, and moving them around such that similar ideas were close to one another and different ideas were far away. Computing diversity might then involve calculating whether selected ideas came from different parts of the table, spanned a large area of the table, *etc.* While different mathematical representations of design spaces and how to quantify their coverage may lead to different definitions of diversity, the central idea remains the same.

The rest of this section first reviews how to represent the space of options — namely, via a similarity function between ideas. Then it presents two existing state-of-the-art methods to compute coverage over that space — one that uses clustering (*i.e.*, buckets) via additive sub-modular functions and one that uses on continuous spaces via Determinantal Point Processes. Lastly, we present additional experiments that compare the conditions under which one diversity measure outperforms the other.

While we selected the below methods to demonstrate our ranking approach for a concrete, real-world example, it is important to note that this work's main contributions — how to combine quality and diversity measures to efficiently compute ideas rankings — do not depend on those specific choices. As we describe in more detail below, our ranking approach (Sec. 4.4.2) applies to any choice of design space representation and diversity coverage measure, provided that they satisfy two mild technical conditions.[10]

[10]In brief, 1) the space must allow one to compute a positive-semidefinite similarity function between points in the space and 2) the diversity function must be *sub-modular* (*i.e.*, obey diminishing marginal

4.4.1.1 Representing Ideas and Computing their Similarity

Before we can compute coverage over space, we need to represent ideas such that we can compute the similarity between them. This is generally done in one of two ways.

The first and most common way is to explicitly represent ideas within a Hilbert space — *i.e.*, a space that permits inner products, such as Euclidean space — and then compute how similar ideas are by taking inner products between them in that space. For example, one can represent geometry or CAD objects using a vector of parameters from a parametric model or using latent semantic dimensions learned from the geometry [63, 282, 45]. For images or sketches, one can use image processing techniques like SIFT features or deep learning (*e.g.*, Sketch-a-Net [280]) to transform free-hand sketches to a vector space. For ideas expressed through text, one can use a bag of words or latent vector space models, such as Latent Semantic Analysis [83]. For mixed-media designs, such as combinations of sketches and text, one can even learn joint vector space models [212]. The similarity is then computed through, for example, cosine, Jaccard, or squared Euclidean distances between those two vectors.

The second way is to compute the similarity between ideas directly using either a *kernel function* — a function that, given two ideas, computes the similarity between — or by having humans directly rate the similarity between ideas [252]. The former is useful in design when one wants to compute diverse, high-quality rankings of structured objects — that is, designs expressed as graphs or hierarchies, such as Function Structures [215] or Function Decompositions [150, 247] using Graph Kernels [269]. The latter is useful when

utility).

ideas are too difficult or complex to easily describe using a set of analytical functions, but one has human experts on-hand who can provide similarity judgments (*e.g.*, idea A is closer to idea B than C, *etc.*) [252]. Through asking human experts (or crowd-sourcing the task), one can compute a kind of "Human Kernel" that can provide sufficient information for our below ranking technique to use.

This work's main contribution — an efficient ranking algorithm for high quality and diverse ideas — is agnostic to the above choice of similarity function. However, a similarity function or matrix, whether chosen analytically or computed by humans, does need to satisfy one mild technical condition — it must be positive-semidefinite. In practice, most widely used methods of computing similarity between vectors, such as cosine, radial basis function, or hamming distances satisfy this condition. If one wants to use their own similarity function, this condition is also straightforward to verify.

For the rest of the chapter, we will assume, without loss of generality, that we can compute a symmetric similarity matrix L whose entries $L_{i,j}$ correspond to the similarity between ideas i and j, where $L_{i,j} = 1$ means that ideas i and j are identical and $L_{i,j} = 0$ means that the ideas are completely dissimilar.

The next two sections 3.2 and 3.3 introduce two existing, competing, state-of-the-art methods[11] for computing diversity with respect to a similarity kernel. Specifically, submodular clustering [171, 173] and Determinantal Point Processes (DPPs) [162], which correspond, respectively, to thinking about coverage over discrete "buckets" versus vol-

[11] As measured with respect to success at a common benchmark task of automatic document summarization (*e.g.*, at the Document Understanding Conference [171, 173]), which require selecting high quality non-redundant sentences to summarize a document.

umes in continuous spaces. In section 4.4.7, we provide additional experiments that characterize the conditions under which one outperforms the other; we found that DPPs were a more robust choice for different problems and we use them for our experimental results later in the chapter.

4.4.1.2 Clustering-based Diversification

One way to think about covering a space of ideas is to think about ideas as falling into different categories, types, clusters, or "buckets." Diversity might then entail promoting adding ideas to empty buckets and penalizing selecting ideas all from one bucket. That is, we wish to model diminishing marginal utility — that adding an idea to a bucket where one already has lots of ideas is not as valuable as adding a (similar quality) idea to an empty bucket.

This is the approach Lin *et al.* [171, 173] use, where they show that many existing diversity methods are instances of a *sub-modular function*. Sub-modular functions are similar to convex functions, but defined over sets rather than the real line. Such functions are designed to model diminishing marginal utility, which is exactly the mathematical property one needs to model diversity [99]. We propose a metric inspired by the diversity reward function used by Lin *et al.* [171] for multi-document summarization, which rewards diversity of a set of items as shown below:

$$Div_1(S) = \sum_{k=1}^{K} \sqrt{\sum_{j \in S \cap P_k} \frac{1}{N \times M} \sum_{i \in P_k} L_{i,j}} \tag{4.1}$$

Here, $V = v_1, ..., v_n$ is the set of all N items in a set. Subset $S \subseteq V = s_1, ..., s_m$

is the selected M items given K clusters. P_i, i = 1,...,K is a partition of the ground set V into separate clusters (*i.e.*, $\cup_i P_i = V$ and the P_is are disjoint). That is, an item can only belong to one cluster. The square root function automatically promotes diversity by rewarding items from clusters which have not yet contributed items.

To understand the above metric, let us take our example, where the collection has three known topics — compost, food festivals, and online web communities. For illustration purposes, consider that adding an idea on one topic introduces a value of "one" into the square root function. Suppose we want to find the diversity of a set of three items. If all items in this set are on compost (*i.e.*, a single cluster), the fitness will be $\sqrt{1+1+1} = \sqrt{3}$, if we have two items covering compost and one on food festivals, the fitness will be $1 + \sqrt{2}$, while if all items cover different topics we will achieve the maximum diversity of magnitude 3. Hence, diverse sets are rewarded by this additive sub-modular function. In Eq. 4.1, the value $\sum_{i \in P_k} L_{i,j}$ implies that items more similar to other items in their cluster (representative items) receive higher reward when added to an empty set. This concept is similar to [38] used in a recommender system, which identifies a set of representative items, one for each cluster.

In general, finding the set of ideas that maximizes Eq. 4.1 is difficult. In fact, it is NP-Hard since it is essentially a combinatorial optimization problem where the value of adding an idea depends on what other ideas one has already added. When solving such problems, a well-known limit due to Feige [94] is that any polynomial-time algorithm can only approximate the solution to Eq. 4.1 up to $1 - \frac{1}{e} \approx 67\%$ of the optimal. However, this is where choosing a sub-modular function for Eq. 4.1 comes in handy. It turns out that greedily maximizing a sub-modular function — *i.e.*, selecting ideas one at a time such

that each choice maximizes Eq. 4.1 as much as possible — is guaranteed to achieve that $1 - \frac{1}{e}$ bound. This makes greedy maximization of Eq. 4.1 the best possible polynomial-time approximation to an otherwise NP-Hard problem. Equation 4.1 uses this property to obtain strong results, and we also leverage similar properties of sub-modular functions later during ranking to create greedy rankings, as well as to improve the convergence of a global optimizer.

A key limitation of using clusters in Equation 4.1 is that we need to know or estimate, which idea belongs to which cluster. In general, we will not know cluster assignments ahead of time and may need to estimate them using different clustering algorithms like K-means [185], Spectral Clustering [196], Affinity Propagation (AP) or domain knowledge. However, as we show in Section 4.4.7, the performance of Eq. 4.1 drastically depends on both the number and accuracy of any clusters. Moreover, ideas may not fall neatly into mutually exclusive buckets. These limitations led us to consider the next approach which does not require explicit clustering but rather considers coverage as a kind of volume measurement over continuous space.

4.4.1.3 Determinantal Point Processes based Diversification

Determinantal Point Processes (DPPs), which arise in quantum physics, are probabilistic models that model the likelihood of selecting a subset of diverse items as the determinant of a kernel matrix. The intuition behind DPPs is that the determinant of L_S roughly corresponds to the volume spanned by the vectors representing the items in V. Points that "cover" the space well should capture a larger volume of the overall space, and thus have a higher probability. Viewed as joint distributions over the binary variables corresponding

to item selection, DPPs essentially capture negative correlations. They have recently been used [162] for set selection problems in machine learning tasks like diverse pose detection and information retrieval [161].

While conceptually simple and fairly straightforward to compute, DPPs suffer from a couple of subtle numerical and optimization issues when used to rank-order items. We review and solve these in Sec. 4.4.2.1, but, briefly, the problems have to do with the sub-modularity and magnitude of the determinant when comparing growing set sizes. Similar to sub-modular functions, one of the main applications of DPP is extractive document summarization, where it provided state-of-art results. As shown by Kulesza *et al.* [160], one of DPPs advantages is that computing marginals, certain conditional probabilities, and sampling can all be done exactly in polynomial time.

For the purposes of modeling real data, the most relevant construction of DPPs is through L-ensembles [39]. An L-ensemble defines a DPP via a positive semi-definite matrix L indexed by the elements of a subset S. The probability of a set S occurring under a DPP is calculated as:

$$Div_2(S) = \frac{det(L_S)}{det(L+I)} \tag{4.2}$$

$L_S \equiv [L_{ij}]_{ij \in S}$ denotes the restriction of L to the entries indexed by elements of S and I is $N \times N$ identity matrix. For any set size, the most diverse subset under a DPP will have maximum likelihood $Div_2(S)$ or equivalently the highest determinant (the denominator can be ignored for maximizing the diversity of fixed set size). As the similarity between two items increases, the probabilities of sets containing both of them decrease. Unlike the previous sub-modular clustering, DPPs only require the similarity kernel ma-

trix L and do not explicitly need clusters to model diversity. This also makes them more flexible, since we only need to provide a valid similarity kernel (*e.g.*, image or shape kernels), rather than an underlying Euclidean space or clusters.

So what does this all mean for a designer? Let us get back to our example earlier in the chapter. If we represent the four ideas as TF-IDF vectors and compute their cosine similarity, we find that the first two ideas related to compost have cosine similarity with each other of 0.61. The similarity between other pairs of ideas is close to zero (< 0.1). This is expected, as the first two ideas are based on compost and have little in common with other ideas that are based on food festivals and online web communities. When we compute the determinant of the sub-matrix for the first two ideas (the numerator in Eq. 4.2), it is ≈ 0.62, whereas for the determinant of first and third idea is ≈ 1. Hence, DPPs (via the numerator in Eq. 4.2) penalize set that contain similar ideas, without requiring us to define any explicit notion of a cluster. This flexibility (plus the strong comparative empirical performance we note in Section 4.4.7) is why we will use DPPs for our ranking algorithms and experiments in the rest of the chapter.

4.4.2 Diverse Ranking of a Set of Items

Thus far, we have compared and analyzed diversity metrics for sets of fixed size. In such cases, a diversity metric like DPPs will give the same value for any permutation of a set since it does not care about the order of the items within the set. This is not desirable for rankings, where users browse sequentially through an ordered list of items up until they reach some (unknown) user-specific limit. This section addresses how to adopt diversity and quality metrics to such cases and compute objective functions over ranked lists (or,

equivalently, permutations over items in a list). To the best of our knowledge, this is the first time DPPs have been extended to such cases, and doing so involves tackling some subtle but important properties of DPPs over growing set sizes.

4.4.2.1 Diverse Ranking of Ordered Sets using Determinantal Point Processes

To extend DPPs to ranked lists, we first need to review some of the geometric intuition behind how the determinant calculations central to DPPs change as we grow the set size. Specifically, we need to look at the determinant of L_S, which is the portion of the similarity kernel (L) formed by the selected items (S). This square matrix grows as we add items to S. Mathematically, its determinant is the product of the eigenvalues of L_S. Geometrically, the magnitude of the determinant is the volume of the $|S|$-dimensional parallelepiped formed by the elements in set S. This implies that adding elements to a set decreases the determinant.

This behavior creates two problems for ranking. First, as we add items to a ranking, the determinants and thus our diversity measure do not have similar length-scales. This means we cannot directly compare or optimize rankings of different length, which matters if we wish to assemble ranked lists in a greedy fashion by progressively adding elements.

To circumvent this problem, we re-define diversity from Eq. 4.2 to Eq. 4.3 below:

$$Div_3(S) = (det(L_S))^{\frac{1}{n}} \tag{4.3}$$

This essentially scales the diversity of a set of size $|S| = n$ by its size. Geometrically, $Div_3(S)$ is proportional to the side length of a n-dimensional cube with the same

volume as the parallelepiped. For a given set-size, n is constant, so maximizing $Div_3(S)$ is equivalent to maximizing $Div_2(S)$. However, mathematically, $Div_3(S)$ is the geometric mean of the eigenvalues of L_S. It represents the central tendency or typical value of the set of eigenvalues via their product.

A second problem with the determinant is that adding the same item to a short list versus a long list can create two issues: (1) Taking the sum of $Div_2(S)$ for a ranked list would not be accurate as items at the beginning of the list will have much larger impact on diversity compared to items down the list. (2) If two almost identical items are placed in the same set, then the determinant quickly collapses to zero (or close to it), introducing numerical errors that make it difficult to compare good versus bad sets on a finite-precision computer. To address this, we use the log-average to measure list fitness for sets of increasing size:

$$Div_R = \sum_{k=1}^{N} \frac{log(det(L_{S(k)}))}{k}$$
$$L_{S(k)} \equiv [L_{ij}]_{ij \in [1,2,..k]} \tag{4.4}$$

The monotonic nature of logs does not change the optimal set, but helps eliminate numerical and discounting errors during the computation of the diversity score.

Despite those computational issues, the determinant's behavior does have a useful side-effect. Because the determinant begins to collapse once the sets start to cover the space (*i.e.*, additional vectors begin to lie close-by to existing vectors), it creates a natural diminishing marginal utility condition where, once we add sufficiently diverse elements, the rankings of further items are not as strongly influenced by item diversity. What this means is that, at some distribution-dependent point in the ranking, items further down the

list can be sorted by quality only, with little to no change in the diversity score for the total ranking. This has a substantial computational benefit because while computing diverse sets is NP-Hard and thus needs to be approximated, at a certain point we can switch over to a much simpler and optimal sorting task to produce the remainder of the ranking.

4.4.2.2 Measuring Quality for Ranked List

The recommended list of items should not only be diverse but also of high-quality. High-quality items ensure that they are relevant to the design problem. While finding the best quality metric for a set of items is still an active area of research, researchers have developed many tractable solutions, including crowd-voting [259], expert opinion [189] or similarity to prior high-quality ideas [13]. Unlike diversity, evaluations of quality are independent, easy to parallel-process, and not combinatorial in nature; this makes estimating quality (comparatively) tractable using existing techniques. We assume that a quality rating is available for every item, or can be estimated (*e.g.*, using our prior work on quality estimation [13]).

Given a quality rating for every item, we need to define the overall quality fitness for a ranked list. For this purpose, we use normalized discounted cumulative gain (nDCG) a standard ranking metric for relevance judgments in ordered lists [138]. It varies from 0 to 1, with 1 representing the ideal ranking sorted by relevance. This metric is commonly used in information retrieval to evaluate the performance of ranked lists by giving more weight to results appearing at the top of list. If k is the maximum number of entities that

can be recommended, then DCG_k is given by:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{4.5}$$

Here rel_i is the relevance of i^{th} item on the list. $IDCG_k$ is defined as the maximum possible (ideal) DCG for a given set of items *i.e.*, when items are sorted by relevance. Hence normalized DCG is given by:

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{4.6}$$

To get an intuitive understanding of $nDCG_k$, consider the following example. Assume that a challenge has 5 items and that we get two lists of 5 items each. Let the relevance ratings be [11, 5, 3, 2, 1] for these items respectively. We normalize these ratings to [1, 0.4, 0.2, 0.1, 0]. Now let us say that List 1 is represented as $[1, 2, 3, 5, 4]$ and List 2 is $[4, 1, 2, 3, 5]$. Using Equation 4.6, DCG_5 for List 1 equals 1.304 and DCG_5 for List 2 equals 0.927. Here, an ideal list will be one where all items are sorted by the quality and $IDCG_5$ is 1.307. Hence, $nDCG_5$ for List 1 is 0.998 while for List 2 is 0.709. Using this metric, List 1 will be a preferred method as it provides more relevant (higher quality) items early on. Hence, we use $nDCG_N(r)$ as our measure of quality for different permutations r of N items.

4.4.3 Optimization

Now that we have ways of comparing the diversity and quality of different ranked lists, our task is to find the 'best' ranking (equivalently, permutation) that trades off diversity and quality. One naïve approach is to equally weigh diversity and quality and then optimize

over the joint objective. However, such an approach is too restrictive since a designer may prefer a ranking that encourages quality more than diversity, or vice versa. Also, in one domain, it is possible that the highest quality ideas are also the most diverse while in another domain, it may happen that one can achieve significant diversity gains by losing almost no overall quality.

It is difficult to unilaterally predict, for every domain, the appropriate trade-off between quality and diversity. Instead we approach ranking as a multi-objective optimization where we generate a entire trade-off front of different rankings — from purely maximum quality rankings to maximally diverse rankings — that allows a designer to choose the extent to which he or she wishes to encourage diversity over quality or compute how much overall quality (if any) he or she might sacrifice to encourage diversity (our below results suggest that such sacrifices are small).

Multi-objective optimization is used widely where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Without additional subjective preference information, all trade-off solutions are considered equally good. Obtaining the trade-off front gives choice to a designer. For example, a designer may choose a highly diverse ranking during early-stage ideation to explore the design space and then later transition to rankings that more heavily weigh quality. Likewise, if a designer wants to ensure a minimum quality threshold among all obtained ranked lists, our approach allows such constraints. As far as we know, our single proposed ranking algorithm is the first to permit such flexibility when comparing and ranking ideas.

At first glance, getting even close to the optimal ranking seems daunting, if not impossible. Not only is the general optimization problem NP-Hard, but the fact that we

have two objectives (diversity and quality) implies that we need to generate not one, but an entire trade-off-front, of solutions. Mathematically, we know that we will have to approximate the optimal solution to this combinatorial problem (if we want to compute it in polynomial time). To do this approximation, we employ a stochastic global optimizer that relaxes the combinatorial problem into a search over real-valued scores. By themselves, such optimizers do not perform well on permutation problems such as ranking; however, due to the careful choice of our diversity scores above, we are able to leverage the properties of sub-modular functions to construct a greedy algorithm that efficiently computes diverse rankings. This substantially accelerates the convergence of the global trade-off-front.

4.4.3.1 Single-objective Greedy Optimization

A ranking optimized for quality can be easily obtained by sorting ideas by quality. Hence, below we explore the more technically challenging task of ranking ideas for maximal diversity. Many diversification methods like Maximum Marginal Relevance [49] use greedy search to obtain a ranked list of diverse items. Likewise, we propose below a greedy algorithm for DPP-based diversity to find a diverse list of items.

1. $A = \emptyset$
2. $A = A \cup \{S_i, S_j\}$ s.t. $[i, j] = \arg\min(L)$
3. **while** $(U \neq \emptyset)$ **do**
4. Pick an item S_i that minimizes $det(L_{A \cup i})$

5. $A = A \cup \{S_i\}$

6. $U = U - S_i$

7. output A

Here, the method greedily adds members to the set by maximizing the probability given by Equation 4.4. Suppose $U = \{1, 2, 3, ..N\}$ is a set of all N items and L is the $N \times N$ similarity kernel matrix. We find a diverse solution by greedily adding items to the empty set to maximize the diversity of the obtained sets of increasing cardinality. As the logarithm of the determinant is sub-modular and monotonic, this greedy algorithm is theoretically guaranteed to provide the best possible polynomial time approximation to the optimal solution. Our experimental results below also demonstrate that this greedy approach to DPPs leads to a higher diversity ranking compared to any random sample and even MMR.

4.4.3.2 Multi-objective Global Optimization

To optimize a permutation of a set of items, we use N continuous variables mapped to a ranked list where each continuous variable $0 \leq x_i \leq 1, i \in N$ is bounded. The permutation is obtained by sorting the variables. To understand the representation, consider the example below. Let us assume that we have a set of 5 items $V = v_1, ..., v_5$. Two possible candidate item score vectors might be x_1=[0.1, 0.3, 0.9, 0.5, 0.8] and x_2=[0.8, 0.2, 0.1, 0.4, 0.0]. On sorting by value, the corresponding ranks for x_1 and x_2 are $r(x_1) = [v_1, v_2, v_5, v_3, v_4]$ and $r(x_2) = [v_5, v_3, v_2, v_4, v_1]$, respectively. By changing the values of x_i, we can obtain any permutation of items. Note that the permutations are not

unique and many x_i's can map to the same permutation.

An ideal set of items should balance diversity and quality. In a classical optimization approach, we could maximize any one of these two objectives directly by finding the best combination of items to recommend, subject to a given metric. For both, however, we need to optimize across multiple, conflicting objectives. This involves finding sets of solutions that represent an optimal trade-off between diversity and quality. We can then use those trade-off solutions to help designers explore and filter possible items.

In practice, one can use any multi-objective optimizer to explore those trade-offs. We chose to use Multi-Objective Evolutionary Algorithms (MOEAs), specifically the NSGA-II algorithm [77]. We generate the initial population randomly with a real-valued gene of length N. The real value indicates the rank relative to other items in the set. The optimizer selects the next generation of the population using a solution's non-dominated rank and distance to the current generation to avoid crowding. Specifically, we use a controlled elitist genetic algorithm [77] with tournament selection, uniform mutation, and crossover.

4.4.4 Results on an Open Innovation Platform

We now demonstrate how the above methods can produce rankings for real-world design ideas. Specifically, we tested the proposed ranking on idea submission from OpenIDEO, an online design community where members design products, services, and experiences to solve broad social problems [101]. We first describe the dataset and then demonstrate how to use our ranking method to produce idea lists that blend quality and diversity.

4.4.4.1 Dataset

On OpenIDEO, each challenge has a problem description and stages — *e.g.*, Inspiration, Concepting, Applause, Refinement, Evaluation, Winning Concepts and Realisation — where the community refines and selects a small subset of winning ideas, many of which get implemented or funded. During the 'Concepting' stage, participants generate and view hundreds to thousands of design ideas; in practice, the number of submissions makes exhaustive review (even of the titles) impossible — *e.g.*, for a medium-sized challenge of ≈ 600 ideas, it would take a person over 25 hours to read all entries.[12]

To demonstrate our multi-objective optimization results on a concrete example, we use a challenge from OpenIDEO entitled *'How might we better connect food production and consumption?'* The Food production challenge had total 606 ideas with a vocabulary size of $1,656$ words and total $88,813$ words after pre-processing. For pre-processing the text data, we use standard natural language processing techniques to convert text to normalized word-frequency vectors (called TF-IDF vectors[83]). Specifically, we use a bag-of-words model to represent items as TF-IDF vectors. For pre-processing, we use Porter stemmer, Wordnet lemmatizer and remove stop-words. All words with inter-document frequency less than 1% and greater than 90% are ignored. We define the similarity between vectors ($L_{i,j}$) by computing the cosine-similarity between the TF-IDF vectors to get the similarity kernel L or any sub-kernel L_S for any subset of ideas $S \subseteq V$.

For any given idea, OpenIDEO has multiple metrics that indicate the quality of

[12] Assuming 200 words per minute at 60% comprehension with the average OpenIDEO idea length of 500 words. This is conservative since many submissions also include images or videos.

an idea: 1) Applause — users can endorse an idea by pressing the 'Applaud' button; 2) Citation count — users can cite ideas that inspired them, similarly to academic papers; 3) Comment or View count — each idea tracks the number of comments or views it receives; 4) a small set of winners proceed to the next stages and win the challenge — those that advance should correlate positively with quality. We use applause as our measure of quality since OpenIDEO uses applause as their own quality measure during Concepting stage. The applause count of any idea i (app_i) is similar to Facebook 'Like' feature, where community members endorse an idea. We did not combine applause with views and comment count metrics as there is no straightforward way to determine optimum weights for combining these metrics. For example, it is difficult to argue if receiving more comments is more important as receiving more views. Secondly, we found that Applause had a Pearson's linear correlation of 0.65 with views and 0.69 with comment count, so choosing a different quality measure does not substantially alter our results. We evaluate our methodology using relevance defined in Equation 4.7.

$$rel_i = \frac{app_i - min(app)}{max(app) - min(app)} \tag{4.7}$$

4.4.4.2 Trade-off between Diversity and Quality

For 606 ideas, the number of possible permutations (*i.e.*, rankings) is $606! \approx 10^{1424}$, which is impossible to compute exhaustively to obtain the ideal trade-off front. We use NSGA-II for bi-objective optimization to simultaneously maximize DCG Applause defined in Eq. 4.6 and Diversity defined in Eq. 4.4.

We use a population size of 500 and run the optimization for 1000 generations with

a crossover rate of 0.8 and mutation rate of 0.01. Greedy solutions for applause and diversity are introduced into the population at first generation to speed up convergence. We get 175 unique points on the trade-off front. The trade-off front between quality and diversity is shown in Fig. 4.1. The values for both objectives are scaled between 0 to 1, with the optimization problem posed as minimization of both objectives. Note that each point on the trade-off front is a permutation of all ideas — that is, each point on the trade-off front represents a different possible ranking (*i.e.*, permutation) of the 606 ideas.

While this trade-off front lets a designer choose different rankings, depending on how much they prefer quality over diversity or vice versa, some designers may want just one ranking of ideas. To achieve this, we propose using indifference curves [65] for selecting an intermediate solution B on the trade-off front. After we normalize the objectives, every circle that uses the origin (*i.e.*, the Utopia or Ideal point) as its circle center can be considered to be a true indifference curve. The points on smaller radius indifference curves are more desirable than those on bigger radius indifference curves. Therefore, the best solution is the point on the frontier that is tangent to the smallest valued indifference curve. In this way, indifference curves essentially weigh diversity and quality equally to provide a single ranking — point B. However, our approach can be easily adapted to different ratios of preferences by altering the shape of the radial curves or even running a one-dimensional search along the trade-off front using techniques like interleaved comparisons [127] or knee region detection[76].

To compare the types of rankings produced by our proposed approach on a concrete example, let us take three points on the trade-off front marked as A, B and C. The maximum quality permutation C sorts ideas by applause while the maximum diversity

permutation A is the one obtained by our above greedy search. We list the top 10 ideas in List A, B and C in Table 4.3. One can notice that solution C (ranked purely by highest applause) has no overlap with most diverse solution A. Reading through the ideas in A (the most diverse ranking), one can notice that despite being diverse, they are poorly written and somewhat irrelevant to the challenge. For example, idea titled "Branded Clothing" proposes referencing local producers on hats and t-shirts. It is a two line idea, without any details on implementation, practicality etc. We found that these ideas often have poor quality scores as they did not address the challenge requirements, were not well written, and did not engage with the community in improving these ideas. Although permutation A is most diverse, suggesting such a set may not be useful for inspiring a designer. In contrast, the permutation C (with the highest quality) has several redundant ideas. The top 10 ideas in C have two similar ideas on mobile applications and multiple similar ideas related to farms. Our selected permutation B, by comparison, incorporates diversity by retaining seven high quality ideas from the most applauded set (C) and introducing three, one of which discusses schools adopting a program to source local food, another one of replacing fences with planted fruit trees, and a third one proposes traveling movie theater with local food. Having such a balanced list of high quality diverse ideas may be used to provide inspiration to designers to come up with designs.

4.4.5 Discussion

Our ranking approach leads to two interesting observations: (1) A small selection of ideas is persistent along the trade-off front, and (2) studying the determinants of lists provides several insights into the nature of diversity and how diverse rankings compare to

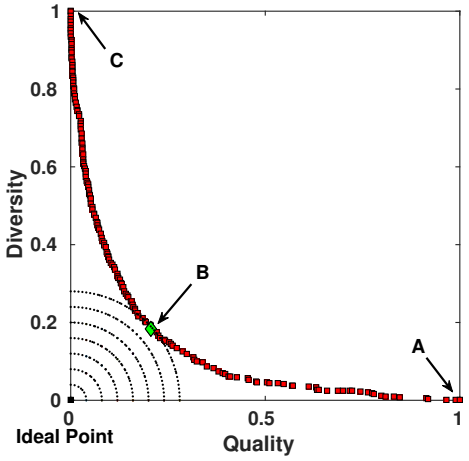

Figure 4.1: Trade-off front between diversity and quality of ranked lists. Each point is a different permutation of 606 *ideas. A is the most diverse solution while C is the solution with the highest quality objective. Indifference curves are used to find the Point B closest to the Ideal Point.*

alternative rankings like highest-quality, MMR, or random permutations.

4.4.5.1 Persistence of Ideas on the Trade-off Front

One key observation is that a small set of ideas persist in the top 10 ranked items across the trade-off front. Taking the top 10 highest ranked items on all 175 lists obtained on our trade-off front, we find that they contain only 36 unique ideas as shown in Fig. 4.2. This means that a designer can read only 6% of the 606 ideas in the challenge, and still get a snapshot of ideas ranging from the highest quality to most diverse. This also aligns with our previous observation in [15], where a small subset of ideas was found to persist on the trade-off front for a different design problem. It is also interesting to note the ideas with very high frequency on the trade-off front like "The Farmer and The Chef". The idea is

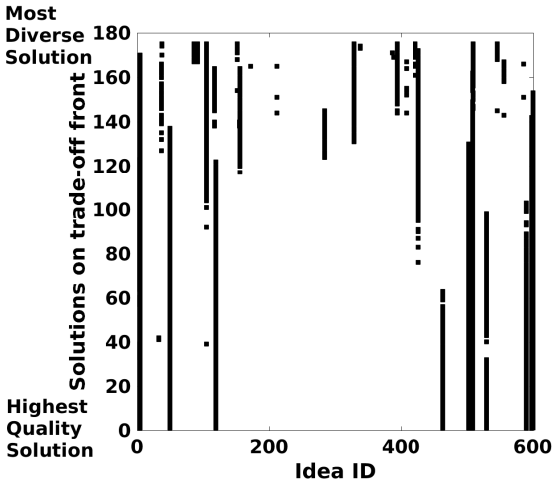

Figure 4.2: Ideas selected in top 10 of different solution sets on the trade-off front between quality and diversity. The figure shows that only a small set of 36 unique ideas appear on trade-off front (the lines in the figures). On the bottom are ideas selected for high quality in the trade-off front, while top of the figure has ideas with high diversity.

both unique and high quality, due to which it is present in top 10 ideas for 97% of the lists on trade-off front. One of this work's ancillary outcomes is to identify such high quality unique ideas.

4.4.5.2 Effect of Diversity with Increase in Set Size

Figure 4.3 shows the determinants for ordered subsets of different permutations. That is, it plots $det(L_{S(k)})$, where as defined before, $L_{S(k)} \equiv [L_{ij}]_{ij \in [1,2,..k]}$, or how the determinant changes as you add ideas from progressively further down the ranked list. It includes the highest quality ranking (C), the most diverse ranking (A), and our intermediate ranking (B). To compare our greedy algorithm with existing methods in the literature, we also plot

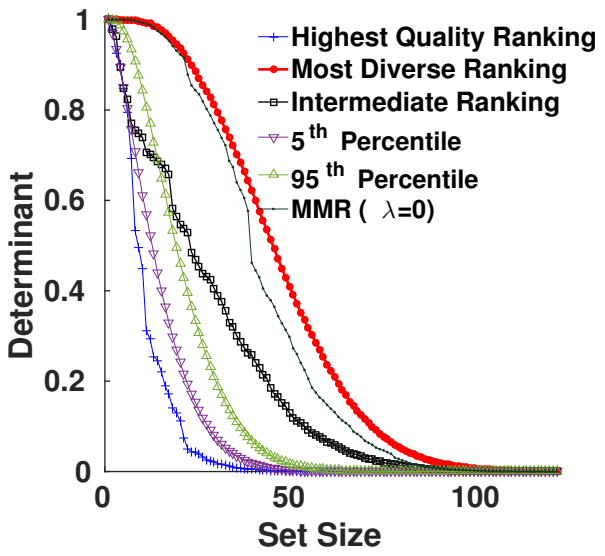

Figure 4.3: Determinant of subsets for different ranked lists. The 5^{th} and 95^{th} percentile solutions show that marginal gain in diversity after 60 solutions is very low. The most diverse solution (A) from trade-off front selected using greedy solution is significantly more diverse than random permutations.

the maximum diversified permutation using MMR [49] with $\lambda = 0$, as well as 5^{th} and 95^{th} percentile from 5000 random permutations to compare to random chance. Figure 4.3 provides four insights into using determinants as diversity metric.

First, Fig. 4.3 shows that our diverse greedy list outperforms both randomized rankings and MMR, in terms of promoting diverse rankings.

Second, We can see that the most applauded set is below the 5^{th} percentile of diverse sets. This shows that, for this challenge, ranking ideas purely by quality produce a ranking that lacks diversity, even compared to random rankings. On the other hand, using the greedy solution to obtain solution A (or even our intermediate solution B) leads to big

gains in diversity, significantly even above the 95^{th} percentile. This indicates that our greedy algorithm is efficiently finding a diverse solution.

Third, the determinants collapse to zero for at most 100 items in the ranked list. This implies that there is not much marginal gain in diversity once one has added many items (*i.e.*, beyond 100) — this makes sense since, by that point, new items will not drastically change the geometric mean of the volume spanned by the determinant. This also allows us to save computational effort by only maximizing Eq. 4.4 up to $N = 100$ and then sorting by quality further down the list. This exact N cutoff will be problem dependent; however, Fig. 4.3 is one criterion for determining when that transition takes places.

Lastly, one can also notice that the determinant magnitude decreases as set size increases. This intuitively makes sense since Eq. 4.4 scales the diversity of sets of different sizes by using geometric mean. Thus, the area under this curve will prioritize diversity in elements early on in the ranking.

4.4.6 Results for Ranking Design Sketches

To demonstrate the applicability of our method to non-text design problems, we take a simple example of ranking five sketches. We adopt the design problem discussed in [235], where one has to sketch a semi-autonomous device to collect golf balls from a playing field and bring them to a storage area. Inspired by the sketches in [235], five sketches for possible devices are sketched by one person, as shown in Fig. 4.4. The sketches are numbered 1 to 5.

To apply our method, we need the quality ratings and similarity kernel for these

sketches. Unlike text ideas, these sketches are not represented as vectors. Hence, we solve a sub-problem of estimating the similarity between sketches using a human rater.

To do so, we decide to learn an embedding of data based on similarity triplets of the form, "Sketch A is more similar to Sketch B than to Sketch C". To find the similarity between these sketches, we ask a human rater to give his relative preferences as shown in Table 4.1. The rater is asked to provide ten comparisons, where he specifies which sketch is closer to the base image. So rating provided in row 1 of Table 4.1 implies that Sketch 3 is more similar to Sketch 1, compared to Sketch 2. Using these triplet ratings, we learn a two-dimensional embedding for all sketches using t-Distributed Stochastic Triplet Embedding (t-STE) [264]. The model is used to obtain a truthful embedding of the underlying data using human judgments on the similarity of objects. Essentially, the model takes as input the triplet embeddings shown in Table 4.1 and generates a lower dimensional vector embedding for each sketch.

Fig. 4.5 shows the output of t-STE model — a two-dimensional embedding for the five sketches. From the embedding, one can conclude that Sketch 1 is quite unique (far away from all other sketches). Using distances from this embedding, we calculate a similarity kernel shown in Fig. 4.6. From the similarity kernel and the two-dimensional embedding, one can notice that the rater found Sketch 3 and 4 similar to each other, while sketch 1, 2 and 5 are relatively unique. Having obtained the positive semi-definite similarity kernel, next we find quality ratings for all the sketches.

We ask a human rater to provide quality ratings for the sketches on a scale of 1 to 10, with 10 being the highest quality idea. The quality rating provided by the rater for these sketches are 3, 2, 7, 8 and 6 respectively. Using these ratings, if we sort these

sketches in descending order of quality, we obtain the following ranking: 4, 3, 5, 1 and 2.

Using the quality ratings and similarity kernel as inputs to our method, we calculate the trade-off front between diversity and quality as shown in Fig. 4.7. There are 17 unique solutions on the trade-off front. We also find the intermediate solution using indifference curves (shown using red marker). Below are the highest quality, highest diversity and the intermediate rankings on trade-off front:

- Ranking by Quality: 4, 3, 5, 1, 2.
- Intermediate Ranking: 4, 5, 2, 1, 3.
- Ranking by Diversity: 2, 5, 1, 4, 3.

From the rankings obtained, one can verify that ranking by quality (left extreme of trade-off front) has sketches sorted by quality ratings. For the most diverse ranking (right extreme of trade-off front), the method gives higher ranking to the unique sketches 2, 5 and 1, followed by similar sketches 4 and 3. Finally, the intermediate ranking balances quality with diversity.

While this example was simple and only 120 permutations were possible for a small set of five sketches, it demonstrated a straightforward way to adapt our method for a sketch based design problem by first estimating the quality and similarity and then generating the trade-off front.

4.4.7 Comparing Diversity Measures

To select the right diversity metric, we compare how accurately the DPP-based $Div_2(S)$ and sub-modular-function-based $Div_1(S)$ metrics capture diversity on a two-dimensional

Sketch A	Sketch B	Sketch C
1	3	2
1	4	2
1	5	2
1	3	4
1	3	5
1	4	5
2	3	4
2	3	5
2	4	5
3	4	5

Table 4.1: Triplet query responses provided by a human rater. For each row, the participant found the item in Sketch A column to be more similar to the item in Sketch B column than the item in Sketch C column.

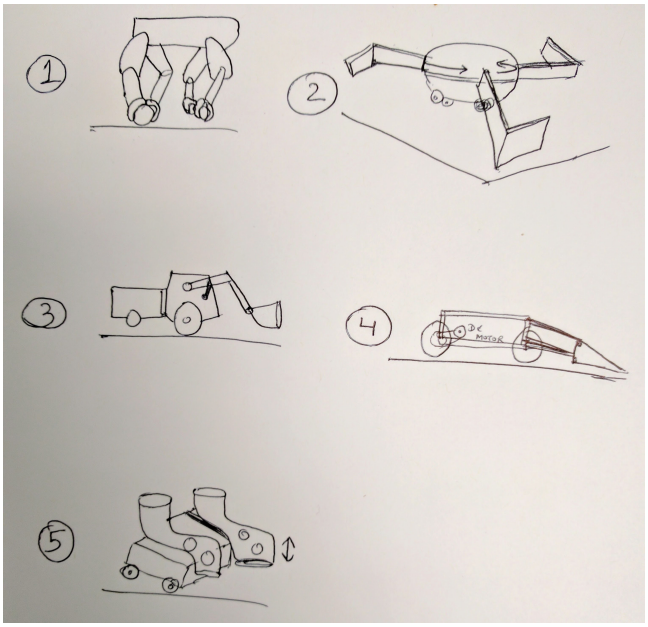

Figure 4.4: Five sketches of semi-autonomous device to collect golf balls from a playing field.

data set, where results can be verified by known ground-truth clusters. This helps us in discussing each method's advantages and disadvantages.

4.4.7.1 Fixed Set Size Comparison

We use an existing clustering dataset shown in Fig. 4.8. It is a two-dimensional dataset with 500 data points across 15 clusters and has traditionally been used to compare clustering algorithms [96]. We use it to compare proposed diversity metrics under the criteria that a set is diverse if it has items from different clusters. This clustering interpretation is widely used in recommender systems for partitioning user profiles [286] and information retrieval for grouping search intents [57]. In Fig. 4.8 each point is allocated to a cluster and the cluster centers are plotted by black square markers.

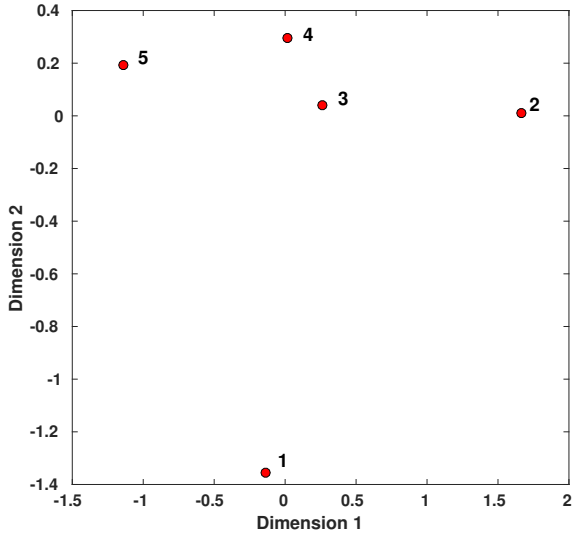

Figure 4.5: Two-dimensional embedding of five sketches calculated using t-Distributed Stochastic Triplet Embedding. It shows sketches 3 *and* 4 *are similar to each other, while* 1, 2 *and* 5 *are unique.*

Suppose we want to select a diverse set of 8 points. Under our criterion, we would prefer to pick points from 8 different clusters; selecting multiple points from the same cluster would be less diverse. Mathematically, this cluster coverage can be quantified using Shannon entropy [141]. Entropy measures the level of impurity in a group and will be maximum when each cluster has same number of elements and will be minimum if a single cluster has all the elements and other sets are empty. We considered a diversity metric 'better' if it provides a higher fitness to a more entropic set (*i.e.*, favors points from different clusters in our gold standard cluster datasets). To assess this, we created two sets of points, Set 1 — high entropy, diverse, plotted using black squares — and Set 2 — lower entropy, less diverse, plotted using red diamond markers. We then compare under what

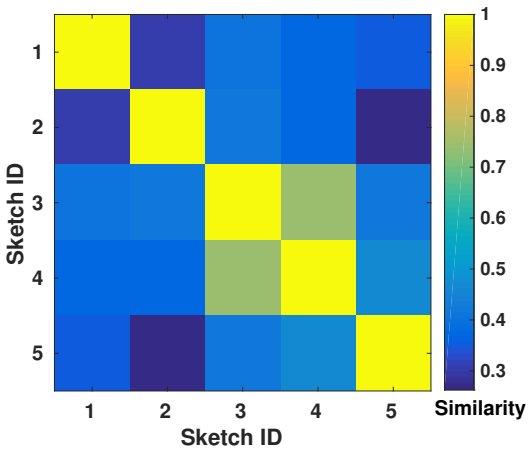

Figure 4.6: Similarity kernel for five sketches calculated for 2-D embedding.

conditions the two methods agree that Set 1 is more diverse than Set 2.

Figure 4.9 compares the above metrics by plotting two set of 8 points each. Set 1 (the sub-modular clustering method) uses black square markers while Set 2 (DPPs) uses red diamond markers. Set 1 is more entropic than Set 2 it has 8 points belonging to 7 unique clusters while Set 2 has 8 points belonging to only 5 unique clusters.

For the DPP similarity measure between points we use a radial basis function (RBF) similarity kernel. This similarity measure used gives score close to 1 to points which are nearby and low scores to distant points. For Eq. 2.2, we need the similarity matrix and the cluster labels for each data point. As a fair comparison, we use the same similarity kernel used for DPPs, but varied the clustering method and number of clusters since this method's performance depends on the clustering labels used for each data point. Specifically, we tested using the already known ground truth cluster labels (*i.e.*, knowing the true clusters ahead of time), and the more realistic condition of computing the clusters using

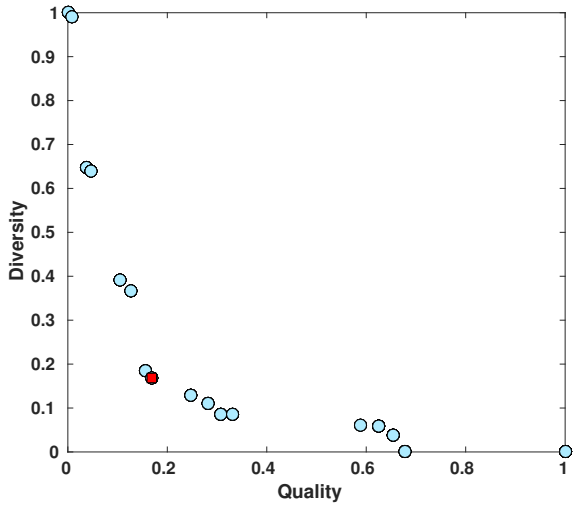

Figure 4.7: The trade-off between Quality and Diversity for Ranking of five sketches.

two methods: Spectral Clustering with 5, 10, 15, or 20 clusters, and Affinity Propagation (AP), which estimates the number of clusters from the data (it estimates 37 clusters for this data set).

When we use the true 15 Gold standard clusters provided with the data set, as expected, the measure agrees with Entropy, which is also defined using the same labels. When we use the similarity matrix defined before and apply Spectral clustering on it for 5, 10, 15 and 20 clusters, the results vary in agreement with entropy. Surprisingly, when the clustering is done with 15 clusters but using Spectral Clustering instead of pre-known clusters, the method finds Set 2 more diverse. We also use Affinity Propagation for clustering, which does not require pre-specifying the number of clusters and it finds 37 clusters in the dataset.

For the DPP metric, we find that $det(L_{Set1}) > det(L_{Set1})$, implying Set 1 more

diverse than Set 2 as shown in Table 4.2. This agrees with our entropy criterion. For sub-modular clustering, its performance was particularly sensitive to number of clusters used. When provided with the true cluster labels, as expected, it agrees with entropy. When it had to estimate the cluster labels, performance varied. Surprisingly, even when told to estimate the correct number of clusters (15), this particular choice of clustering algorithm negatively affected performance. It is possible that a different clustering algorithm (other than Spectral or AP) might offer more robust performance; our point here is that sub-modular clustering is particularly sensitive to how points are clustered and it is not immediately obvious how to verify one has made the "right" choice on a problem with unknown ground truth.

4.4.7.2 Growing Set Size Comparison

How does the above performance difference change if we change the size of the set? Intuitively, if we are given two sets of two points each, it should be easier to estimate which is more diverse compared to when we have 20 points in each set. Figure 4.10 compares DPPs with sub-modular clustering methods as we vary the set size from 2 to 20. We randomly picked 1000 sets of that size and divided those sets into two groups of 500 each. We then conduct 500 comparisons using one item from each group. We calculate the fitness using each method and record how often each methods agrees with entropy (our ground truth measure). Better metrics should agree with entropy more often and should consistently agree as the set size increases. For clarity, we have shown four cases in Fig. 4.10. For sub-modular clustering, using five clusters performs as poor as random chance, while using the known gold standard 15 clusters obtains the best performance, as

expected. The DPP diversity metric performs similar to Sub-modular diversity with 37 clusters found using Affinity Propagation algorithm.

What do this results imply? Given the known clusters, sub-modular clustering has better agreement with our entropy success criterion than those based on DPPs. However, DPPs had more robust performance; that is, if we do not know the exact clusters ahead of time, DPPs perform better on average than sub-modular clustering. In real-world datasets, gold standard cluster labels are rarely available. Even estimating the number of clusters in a collection of design items is difficult. Hence, in such scenarios the parameter-less DPP method is a more robust choice for measuring diversity since using the incorrect number of clusters causes sub-modular-based metrics to perform poorly. However, if a good estimate of number and label assignments for clusters is available, then sub-modular clustering diversity performs well. In the paper, we use DPPs as our diversity metric since we assume that we do not know the number of clusters.

4.4.7.3 Key Assumptions

Below, we list the major assumptions of our work:

1. Our first assumption is that quality ratings for each idea are available. Estimating the quality of an idea is non-trivial and may require expert ratings or crowd evaluation. Incomplete quality ratings (with or without uncertainty bounds) maybe available in many situations.

2. For DPP based ranking, we assume that we can represent all ideas in vector space (or at least find a similarity kernel between them).

Method	Set 1 Fitness	Set 2 Fitness
Unique Clusters	7	5
Entropy	1.91	1.73
DPP	0.0611	1.8509e-04
5 Clusters	0.2201	0.2123
10 Clusters	0.2824	0.3043
15 Clusters (Gold)	0.3289	0.3043
15 Clusters	0.2989	0.3043
20 Clusters	0.3289	0.3043
37 Clusters	0.3289	0.3043

Table 4.2: Objective value of two sets using different diversity metrics.

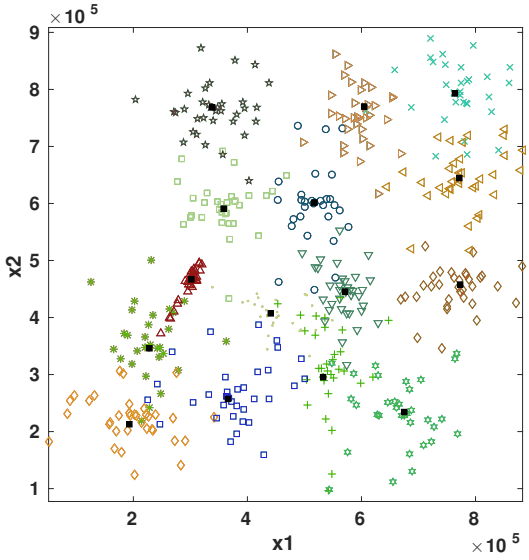

Figure 4.8: Dataset with 500 *points in* 15 *clusters.*

3. For submodular functions, we assume that each item belongs to a cluster and diversity is defined as coverage over all the clusters. We also assume that the clusters do not change.

4. Multi-objective genetic algorithms do not guarantee to reach the optimal solution. Hence, we assume that the trade-off obtained after a fixed number of iterations is sufficient for our application.

4.4.7.4 Limitations and Future Work

We provided a tractable, computational ranking method that simultaneously maximizes a trade-off between quality and diversity of items. As a byproduct, this ranking can also produce diverse, high-quality subsets (such as top 10 lists). However, the method has a

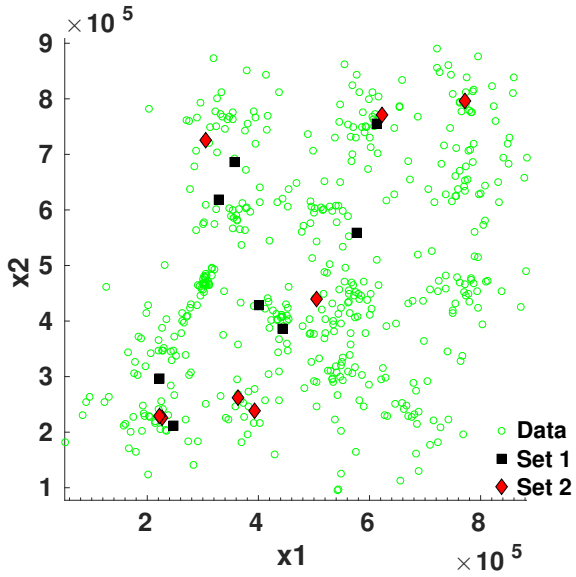

Figure 4.9: Two sets of 8 *points. Set* 1 *is more diverse than Set* 2*, as it has points in* 7 *clusters while Set* 2 *has points in* 5 *clusters.*

few limitations where more research focus is needed.

First, selecting the "correct" diversity kernel to identify similar items is key to the success of any diversification method. At a conceptual level, our main assumption is that the kernel that encodes what makes ideas similar or different is good or accurate. We used a standard cosine similarity kernel for comparing text, however applying machine learning techniques to learn this kernel based on human perception of diversity may improve performance [161]. Also, this method is only able to compute the diversity of ideas within the set of the current data. If all global ideas are considered, the similarity kernel and clustering will change, which will affect the diversity metric evaluations.[13]

[13]To some extent, using humans to construct the diversity kernel may capture this global context, however one open research problem is determining when or for what types of problems that is true.

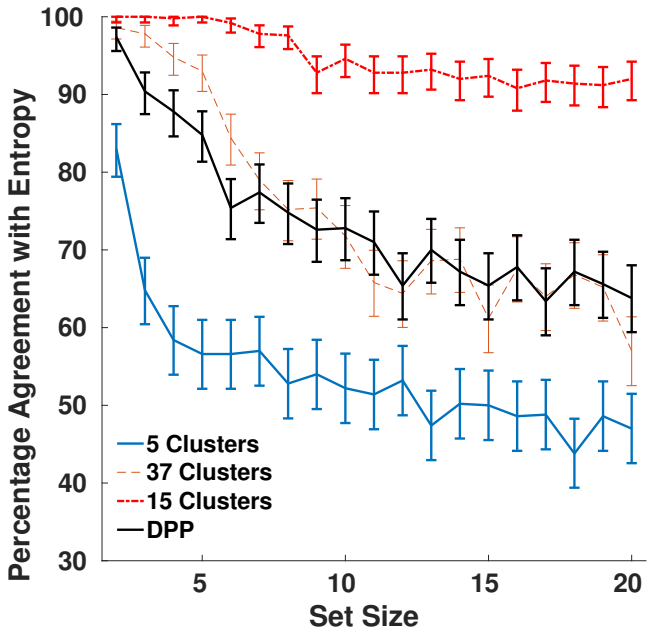

Figure 4.10: Comparison of Sub-modular and DPP Diversity metrics for percentage agreement with Entropy. Random clusters of different sizes are used.

Second, we assume that the high-quality items measured by crowd-voting is desirable for inspiring designers to come up with new designs. The rationale was that items which are more creative and better at addressing the design problem are voted up by the crowd and are good candidates to inspire a designer. This assumption may become invalid if there are other latent factors affecting crowd-voting. However, the main contributions of the work are not really affected by choice of quality metric, since we assume a quality function (however one wants to define it) is available and the contributions are really how to do optimal ranking given such functions.

Third, but related to the second, is that we assume that we have quality estimates for all items. When this is not the case (*i.e.*, the cold-start problem) we would need to

approximate quality by content-based features like item uniqueness. For example, Ahmed *et al.* [15] showed that for OpenIDEO challenges, uniqueness of item and applause are strongly correlated and hence latter can be used in absence of former.

Lastly, our experiment only used text content to represent ideas. This representation was used to facilitate straightforward similarity computation and to demonstrate the key contributions of the work. In real cases, however, many ideas are a combination of text, images and videos, and only computing similarity using text may give an incomplete picture. The proposed method works for design ideas expressed in a variety of ways (text, sketches, function structure graphs, mixed-media, etc.) as all of the important contributions of our method — including how we calculate diversity, the sub-modularity conditions, our greedy approximation, the ranking algorithm, etc. — ultimately only depend on a similarity matrix between ideas (which we called L). If one believes that humans might be the only reliable means to achieve some ground truth understanding of true idea diversity, then this is not a problem for our ranking method; simply use any existing metric- or kernel-learning algorithm to construct L from human evaluators and then apply our ranking method to that new L.

Future research can focus on better methods to compute similarities. For example, one could compute metric spaces over visual designs [63, 282, 45] and combine those with text similarity. In cases where it is difficult or undesirable to compute item features directly, one could use human judgments to compute item similarity (*e.g.*, using techniques like ordinal embedding [136]) and directly substitute this similarity measure into Eq. 4.1 above.

4.4.7.5 Implications for Design Research

Our proposed ranking method applies whenever a designer, team, or decision maker in an organization needs to sift through many ideas. This problem occurs in several design situations: 1) during ideation when multiple designers might generate many hundreds of possible ideas — be they text- or sketch-based ideas; 2) when large organizations wish to gather possible ideas or solutions from employees of their companies, for example via internal innovation tournaments [270]; 3) when companies or designs wish to solicit ideas from crowd-sourcing or online communities; and 4) when a designer wishes to use some kind of computational design synthesis system [54] to generate thousands of possible solutions and then review the output such that he or she understands the scope or diversity of the solutions the system produces. For those above situations, our work has the following implications.

First, our method is the first to enable polynomial time ranking of ideas by both quality and diversity with both provable performance guarantees and flexible control over how much importance the algorithm gives to diversity compared to quality. Such capabilities matter when, for example, designers wish to promote diversity early on in a design process to enable divergent thinking, but then slowly move towards quality convergence over time. Our method provides an easy-to-understand parameter (namely the location along the trade-off front) that allows a designer to adjust how much they care about idea diversity.

Second, our approach provides a concrete metric (namely the difference in the determinant curves in Fig. 4.3) that allows a designer to assess the differences between

the most-diverse and highest-quality rankings, and after how many ideas they have sufficiently covered the available space of ideas. Such observations can provide useful knowledge about a given design problem domain. If our diversity metric plateaus very quickly, it indicates that the domain has very few unique topics. On the other hand, if it plateaus much later, the space of ideas likely has many different topics. Likewise, while not the focus of this work, our method permits a new straightforward comparison of design exploration methods for a given problem; that is, given two methods, by comparing their curves in Fig. 4.3 we can quantitatively study the extent to which different exploration methods cover wider portions of a design space. This allows us to gain new knowledge about both a given design domain as well as different processes designers use to explore it.

Lastly, while our work only addressed trade-offs between quality and diversity, there is no technical reason why our proposed ranking algorithm and methodology could not also incorporate other useful design objectives — *e.g.*, novelty, feasibility, *etc.* — provided such objectives can be evaluated efficiently on a large number of ideas (*e.g.*, via expert or crowd ratings, or using computational evaluation where possible). To enable practitioners deploy this method for their own domain, we have provided the source code [14] and encourage interested readers to use it. To get a trade-off front for any collection of design ideas, a practitioner needs only two inputs — quality ratings for all ideas and a positive semi-definite similarity kernel, showing how similar ideas are to each other. However, the similarity kernel should be chosen carefully, as the diversity is evaluated on the same attributes for which similarity is calculated. For example, let us say a practi-

[14] `https://github.com/IDEALLab/ranking_diversity_jmd_2017`

tioner wants to apply our method to a collection of sketches. Suppose they use similarity kernel based on a surface feature like the color used to sketch the idea. In such a case, the diverse ranking will also output a ranked list, which has sketches of different colors at the top of the list. In contrast, if they use similarity based on some feature like the mechanism used, the ranked list will reflect the same attribute.

4.4.8 Concluding Remarks of Research Task 1

In this task, we proposed a practical, efficient, computational method for ranking diverse and high-quality items. In contrast with past work, we approach idea ranking as a multi-objective optimization problem, which allows a designer to trade off rankings between those that encourage diversity and those that encourage quality. The diverse ranking algorithms can be used for applications like information retrieval, idea filtering or showing exemplars to new participants who wish to work on a new design problem. One open question is, how does diverse ranking affect idea filtering of ideas in practice. Idea filtering may be needed at different points for different processes. The case we discussed in this task assumed that we have a collection of ideas and the quality ratings of all ideas is available. Filtering is used on these ideas to find top-k diverse set of ideas. However, there is another complementary usage of idea filtering where the quality ratings are not available. In many real-world crowd ideation contests (like OpenIDEO) a ranking algorithm is used while the participants dynamically provide the quality ratings (by voting on those ideas). In the next research task, we address this problem, where both the quality rating aggregation and re-ranking happens simultaneously. Our goal is to improve filtering efficiency, that is to find the best ideas in the shortest time.

4.5 Research Task 2: Filtering Innovative Ideas using a Diverse Ranking

In this research task, we investigate how we can apply diverse ranking for filtering ideas. Following successful crowd ideation contests, organizations in search of the "next big thing" are left with hundreds of ideas. Expert-based idea filtering is lengthy and costly; therefore, crowd-based strategies are often employed. Unfortunately, these strategies typically (1) do not separate the mediocre from the excellent, and (2) direct all the attention to certain idea concepts, while others starve. We introduce DBLemons – a crowd-based idea filtering strategy that addresses these issues by (1) asking voters to identify the worst rather than the best ideas using a "bag of lemons" voting approach, and (2) by exposing voters to a wider idea spectrum, thanks to a dynamic diversity-based ranking system balancing idea quality and coverage. We compare DBLemons against two state-of-the-art idea filtering strategies in a real-world setting. Results show that DBLemons is more accurate, less time-consuming, and reduces the idea space in half while still retaining 94% of the top ideas.

4.5.1 Methodology

We first create a dataset of a real-world open innovation problems. In this dataset, we compare the three idea ranking and filtering strategies: i) Majority voting, ii) Bag of Lemons (BoL) and iii) Bag of Lemons with idea diversification (DBLemons).

4.5.1.1 Dataset Creation

Real-world idea selection and summarization: To test open innovation idea filtering we need a dataset containing real-world ideas. A straightforward solution would be to use ideas from an existing open innovation problem and test the new algorithms on the top ideas selected for this problem by experts or by the crowd. This approach has two problems. One, by asking crowd workers to vote on ideas, the text and ratings of which are publicly available, one risks that crowd workers will simply look these ratings up and provide biased evaluations. Two, it is difficult to ascertain if the top ideas selected by the crowd through an existing open innovation platform were selected based solely on merit or if this selection was affected by other factors, like word count and number of comments as previous research indicates [13]. Hence, we decided to generate a new dataset. We proceeded as follows.

First, to make our dataset as close to reality as possible, we gathered a set of ideas posted by community members of a successful online innovation platform, called OpenIDEO. OpenIDEO promotes social impact by designing products, services and experiences that build on the ideas of its distributed community [101]. It hosts idea "challenges" around social issues. Each challenge has four stages: i) Research, ii) Ideas (hundreds of thousands of idea submissions), iii) Evaluation (filtered subset of ideas, 10% of the previous stage), and iv) Winners. To browse through or upvote ideas, OpenIDEO users can order the ideas by date, total number of comments, or total applauds, which are gradually accrued over time. Past work has investigated finding a smaller subset of diverse ideas on OpenIDEO and training classifiers to rank ideas by quality [15] [13]. It

has also been shown that the top-rated ideas in OpenIDEO are in the bottom 5 percentile of diversity, meaning that the top ideas shown are very similar to one another [14].

We created our dataset by summarizing ideas from the Evaluation stage of an OpenIDEO challenge on women's safety[15]. The decision to work with the ideas of the Evaluation, rather than the Ideation stage was taken for two reasons: i) all Evaluation-stage ideas have a minimum quality and structure, compared to ideation stage ideas which may be of poor quality or stubs, ii) summarizing 600 ideas is time-consuming and not central to the research question that is the focus of this work. The chosen challenge called for ideas to solve the following problem:

"How might we make low-income urban areas safer and more empowering for women and girls?"

We summarized each idea in approximately 150 words, taking care to remove identifying information that could lead back to the original OpenIDEO idea description. Each summary was reviewed sequentially by 3 reviewers of the author team to homogenize the writing style and avoid bias due to different writing skill levels. In the end, we acquired a dataset of 52 idea summaries.

Dataset Evaluation: The next step is to evaluate our dataset, and identify the subset of top ideas (hereby called "golden set") that will be used to compare the ranking strategies. Using the judgment-based idea evaluation approach, we use crowd ratings to evaluate each idea. We hired a total of 520 Figure Eight[16] (previously named CrowdFlower) workers and asked each of them to evaluate three ideas on a 5-point Likert scale on the follow-

[15] https://challenges.openideo.com/challenge/womens-safety/refinement

[16] https://www.figure-eight.com/

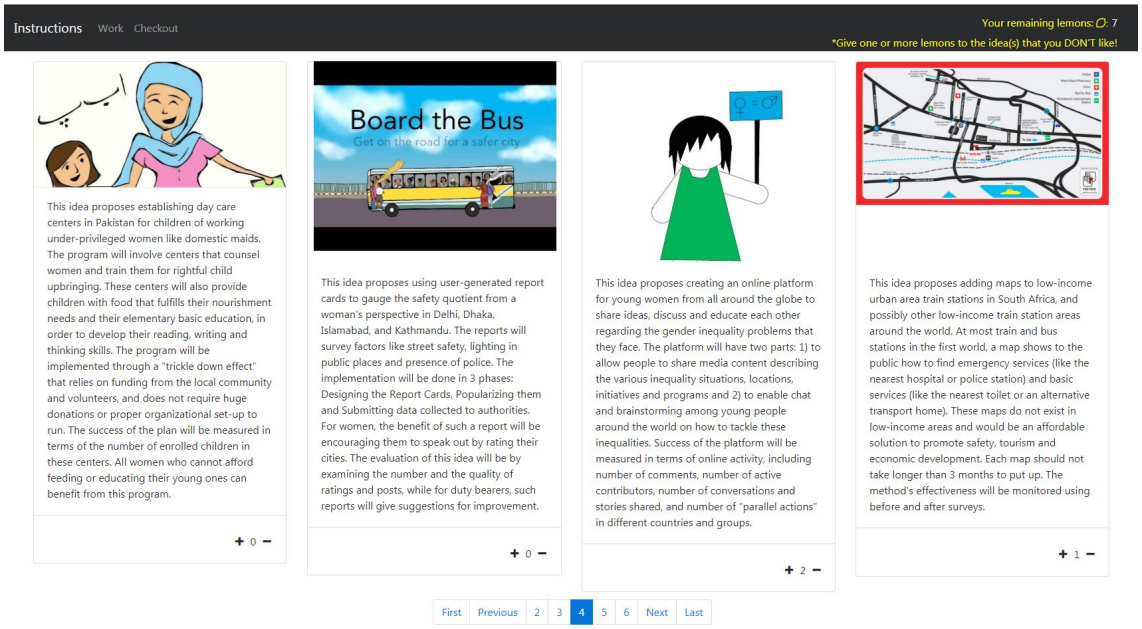

Figure 4.11: Testing platform screenshot, BoL/DBLemons strategies.

ing quality axes: i) Investment potential, ii) Novelty, iii) Impact potential, iv) Feasibility, v) Scalability, vi) Understandability and vii) Overall feeling. The axes were selected in accordance with the common axes used by OpenIDEO to evaluate its ideas in different challenges. In the end, each idea was evaluated by 30 crowd workers. Using the average ratings across axes and workers, we obtained the final quality score for each idea. Using these scores, we selected the top 30% (16 ideas; a similar selection ratio to that of OpenIDEO, which for this challenge selected 15 finalists out of the 52 ideas). These ideas constitute our golden set, over which we will compare the three ranking strategies of our experiments.

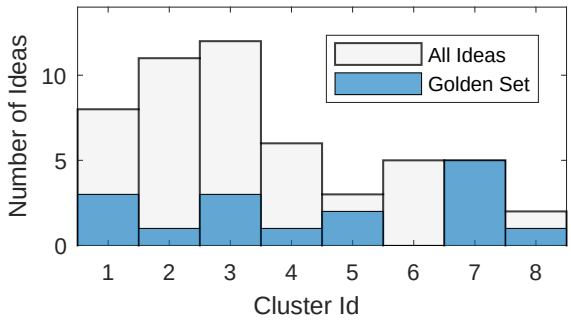

Figure 4.12: Golden set versus total number of ideas.

4.5.1.2 Ranking strategies

Majority Voting: Majority voting replicates the standard voting mechanism used in online design communities. Each rater gets up to 52 votes (the size of our idea dataset). They are free to use them to upvote any number of ideas but they cannot allocate more than one vote per idea. A rater sees the ideas by visiting our idea platform (the functionality of which we present in detail later on, in the Experimental Setup sub-section). When the rater visits our platform, ideas are sorted in descending order by the total number of votes they have already received (i.e. ideas with the most votes go at the top). Dynamic ranking is used, i.e. the number of votes per idea is updated every time a user casts a vote, and this information is used to re-calculate the ranking that a new user sees when he/she first enters the platform.

Bag-of-Lemons (BoL): In this strategy we adopt the Bag of Lemons approach proposed in literature and combine it with dynamic ranking. Each participant is given a budget of 10 "lemons", and they are asked to distribute them to the ideas they feel are the least likely to be selected as winners by an expert committee (the actual winning ideas are

kept secret from the workers until the end of the task). The focus here is on eliminating bad ideas, rather than identifying good ones. Ideas are ranked in ascending order of the total number of lemons they have received (ideas with the least number of lemons, i.e. of higher quality, are at the top). Dynamic ranking is used, in contrast to the Bag of Lemons approach in Klein and Garcia ([154]). The choice of 10 lemons was selected similar to that work for a dataset of similar size (50 ideas in Klein and Garcia ([154] and 52 in our dataset).

Diverse Bag-of-Lemons (DBLemons): In this strategy we combine the notion of diversity with the Bag of Lemons approach. Similarly to the BoL strategy, each participant is given 10 lemons and they are asked to distribute them to the ideas they feel are less likely to win. From the participant's perspective this strategy looks and feels exactly like the BoL strategy. The difference is that after each participant submits their rating, the ideas are ranked by a greedy algorithm, which optimizes for both quality and diversity. Dynamic ranking is used, as in the other two strategies. Idea diversity is calculated using a submodular diversity function, which rewards idea difference (the more different an idea is to the ones already shown to the user, the higher reward it is given by the function). The metric we use to reward idea difference is inspired by the diversity reward function used by Lin *et al.* ([171]) for multi-document summarization. This function rewards diversity and utility of a set S of items as follows:

$$f(S) = \sum_{j \in S} W_j + \lambda \times \sum_{c=1}^{K} \sqrt{|S \cap P_c|} \tag{4.8}$$

Here, $S = s_1, ..., s_m$ is a set of m items (in our case ideas, one idea in this set

denoted by j). The more high-quality ideas and the more diverse ideas set S contains, the higher the value it is attributed by function $f(S)$. Set S is a subset of the original set V of all n ideas (i.e. $S \subseteq V$ where $V = v_1, \cdots, v_n$, and one idea in this set is denoted by i). The first part of the equation controls quality. W denotes the quality vector of the ideas in the set S at a given instance, such that a higher weight implies a better idea. Thus, the higher the quality of ideas set S contains, the higher the value of $f(S)$. The second part of the equation controls diversity. The set V of all ideas is partitioned into k clusters. Each cluster P_c, c = 1,...,k contains a set of thematically similar ideas, and is disjoint from the rest of the clusters (i.e. $\bigcup_{c=1}^{K} P_c = V$ and $\bigcap_{c=1}^{K} P_c = \emptyset$). $|S \cap P_c|$ denotes the cardinality of the subset of S with ideas in cluster k. The square root function automatically promotes diversity by rewarding ideas from clusters that have not yet contributed to set S. Thus, the more ideas from underrepresented clusters that set S has, the higher the value of $f(S)$. Finally, the parameter λ controls the preference given to diversity over quality. A large λ value means that idea diversity will weigh more than quality.

Next, we use a submodular greedy algorithm (Algorithm 5) to order the ideas [195]. Given the set V of all ideas, the algorithm starts with an empty set S. In the end, this set S will be the ranking that the algorithm outputs. It will contain all ideas ordered in such a way as to maximize the objective value defined in Eq. 4.8, i.e. the ideas of high quality and high diversity (i.e. from clusters less represented so far) are at the top of the ranking. To achieve this, the algorithm starts adding ideas to set S and removing them from set V, one idea at a time, such that the selected idea $i \in V$ is the one with the highest marginal gain $\delta f(S \cup i)$ on set S. By choosing at each step to add the idea that will maximize quality and diversity of the existing set of already added ideas, the algorithm not only

selects the ideas but also orders them as well. Finally, as the function in Eq. 4.8 is submodular and monotonic, the algorithm is also theoretically guaranteed to provide the best possible $(1 - \frac{1}{e})$ polynomial-time approximation to the optimal solution.

Algorithm 5: DBLemons ranking algorithm. The algorithm performs a polynomial-time greedy maximization of the gain on the weighted combination between idea quality and diversity (Eq. 4.8). The output is a ranking of all ideas such that high-quality/high-diversity ideas are at the top. Note that this algorithm is same as Alg. 2 discussed before.

Data: Original set V of all ideas

Result: Ranked set S of all ideas

initialization;

$S \leftarrow \emptyset$;

while $V \neq \emptyset$ **do**

    Pick an item V_i that maximizes $\delta f(S \cup i)$;

    $S = S \cup \{V_i\}$;

    $V = V - V_i$;

return S;

Calculating the λ value The DBLemons algorithm can function with all possible values of λ from Eq. 4.8. This means that it can be tuned to favor idea quality or diversity or none, depending on the needs of the ranking. A λ value equal to zero would mean that the algorithm is similar to BoL. Since in this work we examine the effect of diversity, we need to set the value of λ at a high enough value, so that diversity is favored in its output

ranking. As an example, think of the following case. Suppose that in adding a new idea to the ranking the algorithm has to choose between (a) an idea with a high-quality score from a cluster that has already been represented or (b) an idea with a lower quality score from a new cluster (emphasizing diversity). In such cases, we select a value of λ such that DBLemons will always favor diversity and use quality only as a tie-breaker between ideas of the same cluster

To calculate the λ value, which gives the desired diverse ranking, we proceeded as follows. We first take a random uniform distribution of lemons on ideas and then vary the value of λ using a step of 100 from 0 to 10^4 (as we will see, this upper value is more than enough to allow stabilization of the obtained result). We obtained similar results when using other distributions as well, namely the normal distribution, the log-normal distribution and the beta distribution parameterized in five different ways (altering the α and β shape parameters of the distribution, which allowed us to cover a very wide variety of possible vote distributions). Here, for brevity, we report results from the uniform and beta distributions. For each λ value we calculate the rank order of ideas. Then we compare the obtained ranking to the ranking that is acquired from the previous λ value. This provides a measure of how the ranking changes between consecutive values. To compare two different rankings, we use normalized Kendall's tau distance, a correlation metric widely used to compare ranking of items.

To avoid any outlines in the results due to the randomness of the distribution parameterization, we run the experiment 100 times per λ value and average the results. As the λ value increases, the obtained ranking changes, because diversity begins to have an effect on the ranking; however the ranking should stabilize when the λ value is sufficiently

large. Since in this work we examine the effect of diversity on crowd-based idea filtering, we are interested in finding the minimum value of λ after which diversity can fully produce its effects, i.e. the value of λ after which the obtained ranking stabilizes for maximum diversity. Fig. 4.13 illustrates the progress of the mean rank correlation with the progress of the λ value from 100 runs for six different distributions (uniform plus the five beta variations). We observe that for all values above $\lambda = 2000$, the obtained rankings are exactly the same. This gives us an empirical estimate of the minimum value of λ for diverse ranking for the dataset and lemon set size used in this work. Selecting any value above the estimated cut-off will give the same results assuming the true vote distribution is similar to the considered distributions.

Rule of thumb for calculating the λ value: The minimum λ value described above requires a stepwise experiment, like the aforementioned, to identify. For the practitioner or researcher who wants to implement DBLemons inside their idea filtering system, running this experiment may not be feasible, as they may not know the expected distribution of votes. Therefore, below we provide a "rule of thumb" approach for calculating the λ value. Note that this rule of thumb is based on the worst-case scenario, i.e. the extremely unlikely case that all voters give all their lemons to one single idea, and therefore produces a more conservative (i.e. higher) minimum value than the detailed experiment above, which, as we saw, will still produce the same ranking. Nonetheless, its advantage is that it can be easily adapted to different idea dataset, lemon bucket and voter population sizes. We proceed as follows. Our largest cluster has 14 ideas. Therefore the minimum marginal gain of adding a new element is $0.136(\sqrt{14} - \sqrt{13}$ in the square root function part of Eq. 4.8). In our experiments, we use 50 workers giving them 10 lemons each.

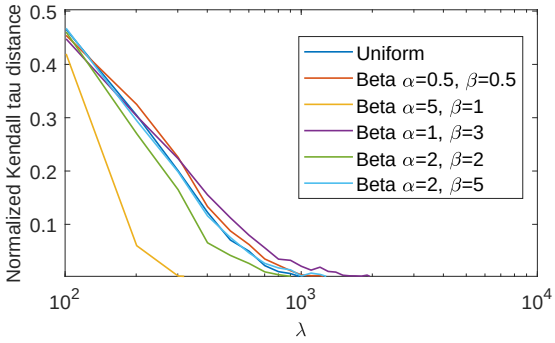

Figure 4.13: Correlation of ranking between successive values of λ for DBLemons measured using Kendall tau distance (log scale for x-axis). After $\lambda = 2000$ the ranking produced by DBLemons remains stable. This is the minimum cut-off value for the algorithm to show a clear preference for diversity.

This means that the maximum possible quality score for one idea can be 500. Hence, to ensure that diversity is always favored above quality, one can simply select any value of $\lambda > (500/0.136) = 3676$. As the true cut-off will always be lower than this value, the rank ordering will be exactly the same irrespective of which value of λ is chosen above this cut-off value. In our experiments, we used a λ value well above the cut-off, namely $\lambda = 10000$.

Idea Clustering The greedy algorithm in Eq. 4.8 requires cluster labels of each idea for function evaluation. Often, these labels are provided by users themselves through tags when posting their idea to an open innovation contest. When this is not the case, one can obtain these labels either manually by placing similar ideas in buckets, or through automatic methods. In our study, we considered two methods of clustering the ideas: (1) automatic text-based and (2) manual concept-based. In the first method, we used the text of each idea to derive its word2vec vector representation, and calculate a similarity score

between each pair of ideas (values in the 0 to 1 range). However, the clustering obtained using this method was unstable with ideas being allotted to different clusters in different clustering runs, mainly due to the fact that there existed little variation between the similarity of the different idea pairs (mean similarity of all ideas was found to be 0.85 and standard deviation equal to 0.04). Hence, we proceeded with a manual clustering of the ideas. We dig deeper into the effect of clustering, and discuss its impact on the performance for our task, as well as for future diversity-based methods, in the section 4.5.3.3.

For the manual clustering two experts, members of the research lab of the author team who had not seen the automatic clustering results, classified the ideas based on their thematic focus. They worked as follows. Each idea was printed in a physical card and spread randomly on a tabletop that served as the collaboration space. The experts then progressively grouped the ideas in thematic groups, moving the cards on the tabletop, through discussion and deliberation. Larger clusters appeared in the beginning, dividing the idea space into a few rough parts, and these were progressively refined to smaller clusters as the experts fine-grained the similarities and differences between the ideas. At the end of this process, eight clusters were identified, with ideas revolving around: 1) *Childcare facilities* (ideas around improving low-income women's opportunities through better childcare support), 2) *Education* (ideas focusing on improving the access to and the quality of training programs for women and girls), 3) *Employment* (ideas focusing on empowering women through novel ways of increasing their monthly income, like community-supported entrepreneurship), 4) *Gender-bias awareness* (ideas focusing on behavioral training of young boys and men on subjects related to inequality and gender-based violence), 5) *Leadership* , (ideas focusing on increasing the leadership potential of

women from low-income areas) 6) *Physical Objects* (ideas involving a physical object to improve the safety of women, like water cleaning systems), 7) *Public Spaces* , (ideas around improving women's safety in public spaces) and 8) *Transportation* (ideas around improving safety within transportation means). Fig. 4.12 shows the number of ideas per cluster and the number of ideas in the golden set.

4.5.2 Results

4.5.2.1 Experimental Setup

Testing platform: To test the three ranking methods, we developed a web platform with a look-and-feel similar to the OpenIDEO platform, with the difference that the ideas are displayed in the ranked order of the experimental strategy that is being used (Fig. 4.11). Ideas are displayed across multiple pages, with each page displaying 10 ideas. To see all 52 ideas, a worker has to go through 6 pages. However, replicating the OpenIDEO functionality, workers are free to evaluate as many ideas as they like, without the need to go through all of the ideas. Each idea is shown in a separate box that contains the idea's image, text summary and an evaluation option. In case majority voting is used, this evaluation option is a "thumbs up" button that the worker can activate (meaning that they upvote the idea), and workers can upvote as many ideas as they like. In case Bag of Lemons or Diversity is used, the evaluation option is a button that adds a number of "lemons" to the idea. Each worker starts with exactly 10 lemons, and they can allocate any number of these lemons to any idea. On the top right corner of the platform we placed a short instructions message, reminding the worker to vote for their preferred ideas (major-

ity voting), or reminding them their number of unassigned lemons (BoL or DBLemons). Workers can change their evaluations (thumbs up or lemon assignments) as many times as they like, until they are satisfied with the result.

Crowd workers: We hired 150 Figure Eight workers (different than the ones used for the idea dataset creation described in the previous), and divided them randomly into the three experimental conditions (one condition for each strategy, 50 workers per condition). No specific test was required for workers to register for the Figure Eight task that gave access to our platform, since we aimed for the typical Internet user, like the ones that usually vote in open innovation contests. Nevertheless, to ensure a minimum guarantee of task attention we opted for hiring mid-experienced but not over-specialized Figure Eight workers (Level 2 out of 3). Each worker was given a link to the platform, and they were paid once they finished their evaluation. As noted above, similarly to real-world open innovation platforms, each worker was free to spend as much or as little time as they wanted reading and evaluating some or all of the ideas of the challenge. Nevertheless, to avoid bias in the results from possible spammers, we cleared the results of those workers that did not rate any idea or did not give any lemon (less than 2% of the hired workers) and replaced them with other workers to reach the desired number of hires (50 workers per compared strategy). Payment was calculated on the basis of \$12/hour[17], and for an average estimated task duration of 15 minutes[18], i.e. \$3 per worker. To further incentivize workers into making as qualitative evaluation as possible, we notified them that the Top 3

[17] `http://guidelines.wearedynamo.org`

[18] As we will see in the results analysis, this time estimation indeed included the average task durations of all three algorithms.

accurate raters would get an additional bonus of $1.

Ranked order calculation: As soon as all workers had finished voting, we obtained the ranked order for each of the three voting strategies. For the Majority voting case, ideas were ordered in descending number of total votes. For the BoL and DBLemons cases, ideas we ordered in ascending number of total lemons. In case of ties, we applied the dense ranking allocation model ("1223 ranking"), according to which when two ideas have the same number of upvotes or lemons, these ideas are given the same ranking place number (e.g. place number 2), and the idea(s) after them receive the next ordinal number (place number 3 in our example). To break ties within the same ranking place, we ordered ideas alphabetically.

4.5.2.2 Performance

DBLemons outperforms BoL and Majority voting in filtering efficiency: Fig. 4.14 compares the filtering performance of the three strategies. We first show the final ROC curve, followed by filtering efficiency curves by using (b) 20 and (c) all 50 voters per strategy. From the ROC curve, it is evident that DBLemons outperforms both BoL and Majority voting with a higher AUC ($AUC_{Majority} = 0.648$, $AUC_{BoL} = 0.671$, $AUC_{DBLemons} = 0.869$). DBLemons also achieves a True Positive Rate (TPR) of 1, with False Positive Rate (FPR) of only 0.44. The other two methods perform similarly to one another, achieving their maximum TPR at FPR of 0.77 and 0.75.

Going a bit deeper into our analysis we observe Fig. 4.14 (b) and (c). Here, the y-axis corresponds to the percentage of golden set ideas identified, using the x% first

ideas of the ranked order of each strategy (x-axis). For example in Fig. 4.14, we see that considering all available voters and using the 50% first ideas of the returned ranked order, majority voting manages to retrieve approximately 60% of the golden set, BoL has exactly the same result, while DBLemons retrieves 94% of the golden set. The dashed vertical line corresponds to the cardinality of the golden set ($16/52 \approx 31\%$ of the total number of ideas). Two main observations can be made using this figure. First, the filtering efficiency of DBLemons is higher than both the BoL and Majority voting strategies, even from the first few voters (Fig. 4.14 b)). When all voters are used, DBLemons manages to identify three quarters (75%) of the golden set using approximately as many ideas as the golden set itself (35% of the ideas when the golden set represents 33% of the ideas). On the other hand, using the same percentage of ranked ideas, majority voting manages to find only 50% of the winning ideas. BoL is left even further behind as it starts making distinctions among ideas using 40% of its ranked order and above. More important, DBLemons manages to identify 94% of the winner ideas using only half (50%) of the idea space, while BoL needs to explore 70% and Majority voting 80% of the idea space respectively.

As we will see in the Discussion section 4.5.3 that will follow, the $20 - 30\%$ *reduction of the idea space size achieved by the proposed method compared to the other two alternatives, means significant gains in terms of effort and cost, and considerably improves the prospects of open innovation adoption by large commercial players.*

DBLemons has a higher distinction ability than the other two strategies: The second remark that we can make concerns the distinction ability of the three strategies. As it can be expected, the fewer people have voted, the more ties we have among the ranked ideas,

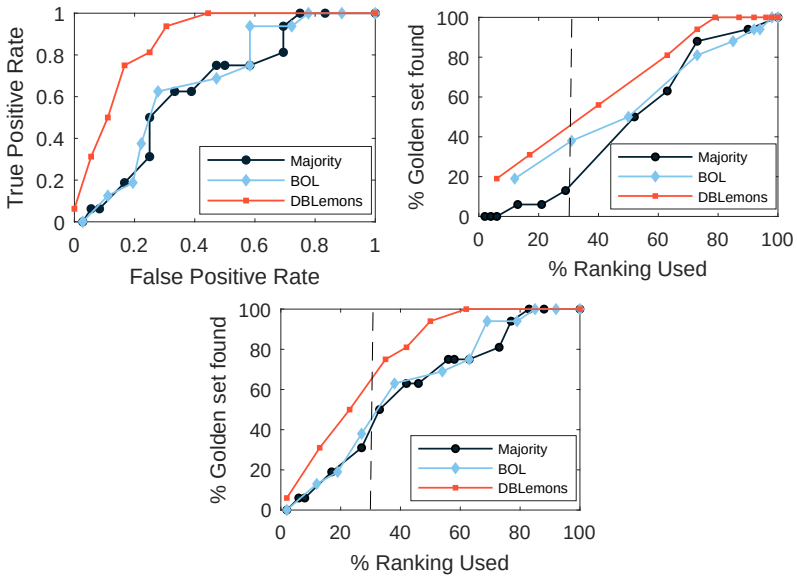

Figure 4.14: Performance comparison of the three ranking strategies. The dashed line shows the golden set cardinality cut-off. DBLemons finds more winning ideas, earlier on, with less workers, and using a smaller portion of the ranked idea space.

and therefore the less distinction capability the algorithms have. This is more clearly reflected in Fig. 4.16, which visualizes the ranking changes per algorithm as the number of participating voters increases. Colored boxes in this figure represent golden set ideas, and blocks of boxes represent ideas that have received the same number of votes/lemons. For example, on the far left column, we see that using a limited (20) number of voters the Majority voting strategy distinguishes 11 total blocks of, i.e. ranks the 52 total ideas with 11 ranked places. To surpass the size of the golden set, and thus give a conclusive answer about ideas can be included in the top 30% winning ideas, it needs a total of 26 ideas (blocks 1-7), out of the total 52. This means that for this specific number of voters, the distinction capacity of majority voting starts from 50% of the idea space. Coming back to Fig. 4.14 (b) we see this result quantitatively: indeed the point of the Majority voting line

that surpasses the vertical 30% cutoff line starts at 50% of the x-axis. Still on Fig. 4.14 (and confirming visually with Fig. 4.16) we can observe that *DBLemons can distinguish enough ideas to reach the golden set size earlier on than the other two algorithms*. For a few voters its distinction capacity starts at 40% of the idea space and surpasses the other two algorithms by 10%, while for 50 voters it requires only 35% of the idea space and is matched only by Majority voting.

BoL and Majority perform similarly in most cases: A last result in terms of performance concerns the comparison between BoL and Majority voting. We observe that when the 20 first voters are employed (Fig. 4.14) and less than 60% of the ranked idea space is used, BoL performs better than Majority voting, finding 50% of the golden set. This result is consistent with [154], which also finds that Bag of Lemons performs better than Majority voting. However, as the percentage of the ranked idea space used by the strategies increases above 60%, the performance of BoL drops comparatively to Majority voting, and the two strategies perform similarly. This similar behavior becomes more apparent when more (=50) voters are employed (Fig. 4.14), in which case the performance of the two strategies is very similar from the start. Although this result does not contradict with [154] due to the different baselines used, it opens up new questions on the applicability of BoL. BoL is reported to perform better in idea filtering when compared to Likert scale voting, while BoL with dynamic ranking is shown to perform similar to Majority voting in most cases of our experiments as seen above.

4.5.2.3 Time on task

Although DBLemons provides better filtering accuracy, one may think that workers take more time in that method. However, we found that both DBLemons and BoL took 42% less median time compared to majority voting (Fig. 4.15). There are two possible explanations for the difference in timing. Prior literature [154] has argued that BoL uses less time as it only requires people to identify ideas that are clearly deficient with respect to at least one criterion. This can be a reason for the lesser time taken by both methods using lemons. However, we also observed that the total number of lemons allocated to the ideas is 43% of the total votes in majority voting. This may imply that workers took similar time per vote (or lemon allocation) in all three conditions. We tested this theory by observing time on task for workers using less lemons. One would expect that workers who allocate less lemons take proportionally less time. However, we did not find clear evidence of this from our dataset, as most workers used all ten lemons. We believe that the less time taken by the lemon-based methods can be attributed to a combination of both factors.

Summarizing, in answer to the two research questions of our related literature analysis, our results show that:

- Bag of Lemons with dynamic ranking (BoL) does not outperform Majority voting in terms of filtering efficiency; however it is speedier in finding good ideas, confirming the analysis by Klein and Garcia ([154]).

- Diversity-assisted Bag of Lemons (DBLemons) does increase the BoL efficiency in

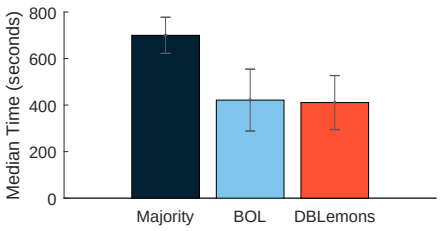

Figure 4.15: Median task times for each strategy.

filtering high-quality ideas.

4.5.3 Discussion

4.5.3.1 Impact on Open Innovation

Overall, our results give rise to two main observations, which impact open innovation:

- Majority voting, used widely by open innovation platforms, is more time consuming and less accurate than the other two alternatives.
- Diversification helps in the selection of high-quality ideas, more accurately than the other two alternatives and in less or equal time.

In specific, we found that BoL is speedier in finding good ideas than Majority voting (as also shown in [154]); however neither BoL nor Majority voting are very accurate. BoL is capable of detecting only 50% of the golden set ideas using 40% of the ranked ideas. Although, this is better than random chance, it is of less practical use. To achieve about 95% accuracy, BoL and Majority voting offer only a 20%-30% reduction of the idea space. In comparison, the DBLemons algorithm captures 94% of the winning ideas with 50% reduction of the idea space and 100% of the winning ideas with a 35% reduction.

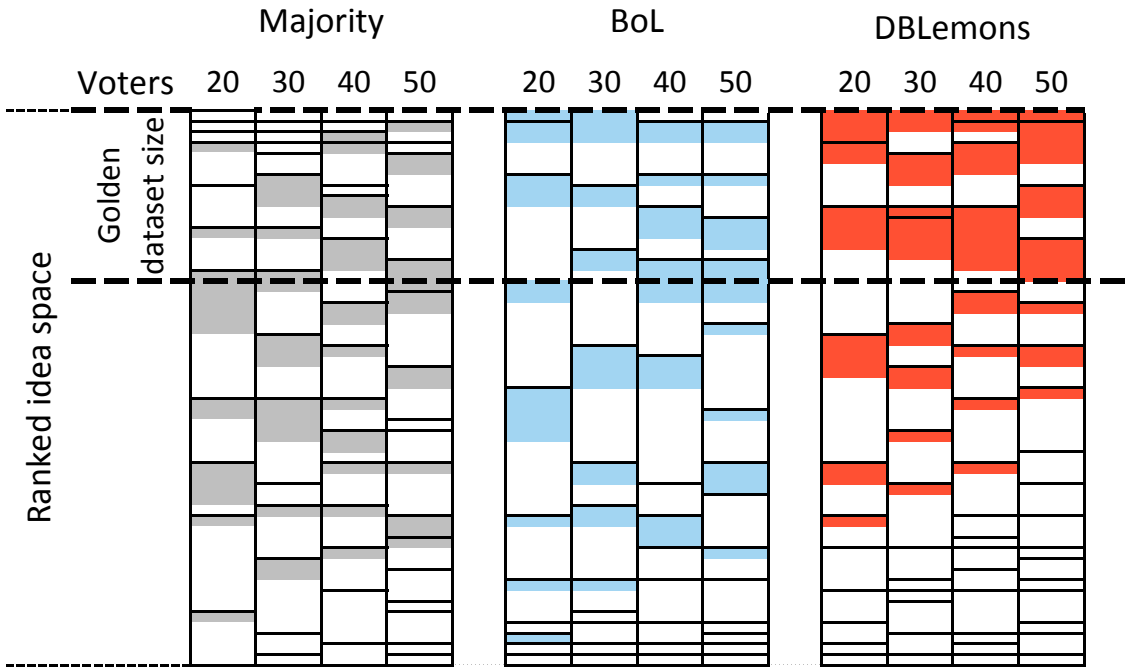

Figure 4.16: Progressive ranking per strategy, descending quality order. All strategies improve with the number of voters, but DBLemons does so faster.

Return-On-Investment: In practical terms, our results show that *with DBLemons a person has to look only at half of the actual ideas to get almost all of the winners*, and that DBLemons also saves this person 20% and 30% of evaluation effort compared to BoL and Majority voting respectively[19]. Assuming that a company wants its experts to evaluate an idea space that includes almost all (95%) of high-potential (golden set) ideas, these experts would have to go through 20% and 30% more ideas if the BoL and Majority strategies respectively were to be used. Considering: i) a median estimate of approximately 100K ideas received per large-firm ideation platform[20], ii) an average of 10% of these ideas reaching the latter stages of the innovation contest (as it is the case with OpenIdeo for example) and iii) an estimated expert cost of $500 and four hours to evaluate one idea in a Fortune 100 company [221], DBlemons can save organizations between $1 and $1.5M in costs and 50-75 person-months in effort. This constitutes a significant Return-on-Investment not only for the proposed method, but most importantly for the prospect of open innovation to be considered as a viable complement to in-house innovation by large corporate players.

4.5.3.2 How diversity helps

A key question is why diverse ranking works so well, and under which circumstances it is expected to do so. We considered three possible explanations. First, it may be that the actual set of golden ideas have equal representation across all clusters. By enforcing an

[19]Evaluation effort is measured in terms of the idea space percentage that a person has to go through after crowd-based filtering.

[20]From [154]: IBM's "Innovation Jam" gathered 46,000 ideas, Dell's IdeaStorm 20,000 ideas, Google's 10 to the 100th project over 150,000 ideas, and Singapore Thinathon's contest over 454,000 ideas.

equal number of good ideas from each cluster on the first page, DBLemons would increase the chance of getting a similar distribution. However, as it can be seen in Fig. 4.12 this is not the case; ideas are represented unequally and disproportionately to the cluster size, across the clusters. Hence, this potentially disadvantages DBLemons by pushing high-quality ideas from same cluster down the ranking. Another reason can be that using lemons instead of upvotes helps in idea filtering. However, by comparing BoL with Majority voting, we saw that lemons by themselves are not more effective than upvotes, although they do help in reducing time on the voting task.

Finally, we argue that DBLemons works better as it provides better coverage of the idea space. We observe Fig. 4.17 (a) depicting the median idea clusters shown by the three algorithms on the first page. Each circle depicts the presence of an idea cluster on the page, and size is proportional to the idea percentage the cluster occupies. Whereas DBLemons always represents all 8 clusters, majority voting shows ideas from only half of the available clusters, while BoL omits two. A similar pattern is repeated across the rest of the pages. Focusing on the behavior of the DBLemons algorithm, Fig. 4.17 (b) illustrates the clusters seen by each worker when entering the platform and using DBLemons.

The y-axis shows individual workers and it has 50 values, one for each worker. The x-axis depicts the ranking of the 52 ideas seen by each worker. Every 10 ideas (values of the x-axis) represent one consecutive page of the ranking. In other words, this figure contains 50 horizontal slices, with each slice representing the idea ranking seen by one individual worker. The color for each idea represents the cluster that the idea belongs, and we have 8 colors/clusters. This color coding allows us to observe that the left part of the figure has successive items with different colors. This means that all the workers saw

ideas from different clusters at the start of their ranked list. We also observe that colors alternate significantly until approximately idea 34 (x-axis value equal to 34). This means that for the first few pages (until page 4 out of 6) the algorithm represented all 8 clusters in equal proportions for all workers. This is indeed the expected behavior of DBLemons ranking, as observed in Figure 4.17. Finally we observe that from the middle of page 4 (x-axis value equal to 35) and until the end of the ranking (end of the x-axis) the colors are similar. This reflects the fact that the ideas that belong to both a large cluster and have received many lemons (i.e. are of low quality), are correctly pushed by the algorithm towards the end of the list.

Essentially, DBLemons tries to maximize coverage over clusters, and in doing so it gives voters an overview of the full idea space right from the beginning. As people visit more pages, they realize that they have already seen similar ideas and can make faster and more informed comparisons, but also to go back and correct their previous evaluations. In contrast, people who see few concepts initially may get fixated on them. This observation is supported by literature on fixation, which proposes that the solution search process should begin with a divergent step prior to convergence [178]. Smith *et al.* ([241]) referred to fixation as something that blocks or impedes the successful completion of cognitive operations (like remembering, solving problems and generating creative ideas), and proved that one's recent experience can lead to unintentional conformity or fixation to a few ideas. The fixation effect is further confirmed by looking at the workers' navigation behavior (Fig. 4.17 (c)). As we can observe, the volume of page visits for the DBLemons is approximately four times higher than that of Majority voting and two times higher than that BoL for the first page, while for the rest of the pages the DBLemons volume is twice

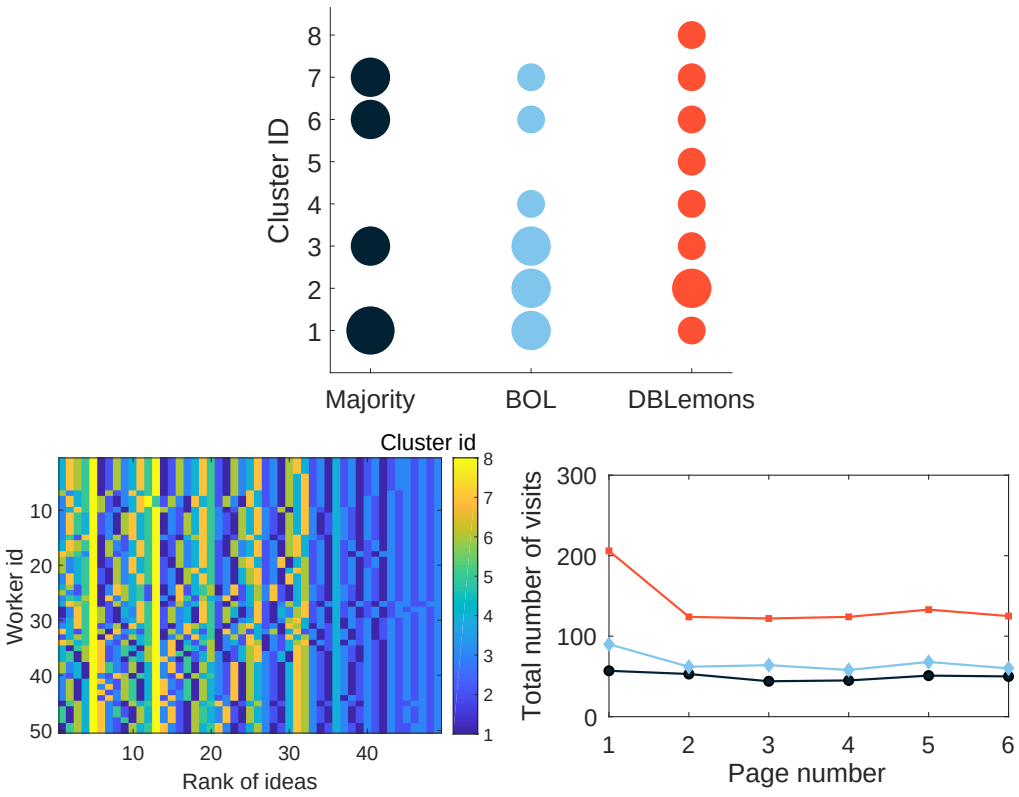

Figure 4.17: DBLemons provides a more even coverage of all idea clusters: (a) Top right: Median cluster entropy, and (b) Top right: Cluster distribution for ranking seen by successive workers. This could lead voters to make more idea comparisons, generating (c) more page visits.

as much than that of the other two strategies. This higher volume, which is in the dataset due to multiple back and forth hops across the pages, supports the possibility that after getting a diverse summary of ideas from the first page, users made comparative decisions regarding their lemon allocation.

4.5.3.3 The effect of clustering

Our work examines the effect of diversification on crowd-based idea filtering. As such, we have maximized the coverage of clusters and obtained them through manual label-

ing, involving two collaborating experts, to minimize noise. Manual labeling can be also obtained through the idea authors, who often categorize their proposed ideas upon submitting them to an open innovation contest (like in the case of our generalization experiment with the UNESCO dataset), or it can be crowdsourced to multiple independent evaluators (in which case elements like inter-coder reliability need also to be examined). However, one may wonder what happens in case the cluster labels are noisy, an issue frequently encountered when fully automatic clustering is employed. To check this behavior, we ran an additional experiment using automated clustering.

We represented each idea as a vector, using Google's publicly available pre-trained word embeddings[21], and then summing these to obtain the sentence embeddings of the idea. This is a widely used method [124], which has also been adopted by the Semantic Textual Similarity shared task [53]. Next we used cosine similarity to compute the similarity between all pairs of ideas, a process which revealed that automatic similarity calculation could not differentiate much between the ideas (mean similarity 0.85, standard deviation 0.04, in the 0 to 1 range). On the obtained similarity matrix we applied spectral clustering with 8 clusters (same as manual clustering) to find the cluster labels for each idea. In doing so, we noticed that different spectral clustering runs provided different clustering assignments, i.e. that the dataset did not have well-defined automatically-identifiable clusters. To further ascertain this, we ran 100 clustering runs and calculated the silhouette score for each run. This score measures how well-defined the identified clusters are and is widely used in literature [28] to judge different clusterings. Silhouette

[21] `https://code.google.com/archive/p/word2vec/`, providing 300 dimensional vectors for 3 million word embeddings, pre-trained on a Google News corpus of about 100 billion words.

coefficients near a value equal to +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters, and negative values indicate that those samples might have been assigned to the wrong clusters. The obtained scores varied between 0 to 0.1, showing poor clustering on the dataset. We nonetheless proceeded with our further analysis, selecting the clustering assignment with the maximum silhouette score. Interestingly, this assignment also had the most – albeit still quite low – similarity with the manual labeling (maximum adjusted rand score 0.29), compared to the other assignments.

Using this automatic clustering assignment, we replicated the DBLemons experiment with 50 workers. We found that this method required 60% of the idea space to find 81% of the winning ideas (ROC $AUC_{DBLemons-auto} = 0.741$). This result is poorer than the original DBLemons, which only required 45% of the idea space for finding the same number of top ideas ($AUC_{DBLemons-manual} = 0.869$), but it is still slightly better than the BoL and Majority voting methods, which required 70% ($AUC_{BoL} = 0.671$) and 80% ($AUC_{Majority} = 0.648$) of the idea space respectively.

To cover the case that this result is due to the particular automatic clustering method used, we also compared four additional clustering algorithms — KMeans, Gaussian mixture models, Ward Agglomerative and Affinity Propagation clustering. For each method, we conducted 100 runs and calculated the maximum silhouette score. Adding these algorithms allowed us to also experiment with fixed defined (8 clusters for the KMeans, Gaussian mixture, and Ward Agglomerative methods) versus non-fixed number of clusters (in Affinity Propagation method). Results with all these algorithms showed that automatic clustering, on the particular dataset, gives noisy cluster assignments with large

differences across runs. The maximum silhouette scores (from 100 runs per algorithm) for these methods were 0.06, 0.04, 0.11 and 0.03 respectively. The values near 0 show that none of the methods was able to give good/decisive results from run to run in regards to the clustering, and none of the methods had significant advantage compared to the others. We also compared all methods (spectral clustering and the additional four above) using a second cluster fitness metric, namely the Dunn score [89]. In this case too, no major difference or advantage was observed in the clustering capability of the algorithms. This noisy clustering result, across different clustering methods, can be attributed to the small size of text per idea, which does not allow the creation of a representative context corpus. In case the algorithms could be provided with context knowledge or in case the text per idea was longer, automatic clustering may have given better results.

Concluding, the relatively poor performance of automated clustering compared to manual clustering was not surprising, since the automated method did not differentiate much between ideas and the idea vector space lacked well-defined clusters. This noise meant that automatic clustering risked placing ideas that are very different from one another into the same cluster, a fact which we also manually verified (see for example Table 4.4). Hence, we observe that the effectiveness of DBLemons partially depends on idea clustering. We believe that this is true for any method that tries to leverage the power of diversity. If noisy labels are assigned to the ideas, any method maximizing coverage over these labels will also be affected, therefore a practitioner must take special care in preparing the idea clusters or defining their similarity. Nevertheless, the fact that even with noisy input DBLemons still manages to obtain better results than the compared alternatives, indicates that the potential of diversity when combined with crowd evaluations is important,

and will continue improving as better clustering methods appear by independent research.

4.5.3.4 Generalization of the results

Another important topic for discussion is the generalizability of the proposed approach. The results elaborated so far are based on the dataset derived from OpenIdeo, because OpenIdeo is among the leading platforms for crowd ideation. For completeness purposes however, it is important to test the proposed approach on another ideation dataset, preferably one from a different platform and challenge context, and one subjected to as little processing as possible in regards to the creation of the golden dataset, the clustering of the ideas, or the idea text creation. The dataset we worked with comes from the 2017 Youth Citizen Entrepreneurship competition of UNESCO's Global Action Programme on Education for Sustainable Development[22]. The competition called for innovative ideas and projects to address important social, economic, environmental, health and governance challenges of our times. The 176 ideas of the dataset were already categorized into one or more of 17 thematic clusters, according to the Sustainable Development Goal that they work towards solving. We used the same clusters, in order to avoid any potential bias from the manual clustering approach used for the original OpenIDEO dataset. In case an idea belonged to two clusters, we used the cluster marked by the idea authors as primary. We also did not intervene in the creation of the golden dataset, which was taken directly from the competition data based on the number of comments per idea. Using the number of comments as a quality indicator was supported by the fact that the ideas with the highest number of comments were also those nominated by the competition panel of judges as

[22] https://www.entrepreneurship-campus.org/ideas/12/

the winners. Finally, to avoid any potential writing style bias from summarizing the ideas, the text that we used per idea was the original ”Explain your idea in details” paragraph included in each idea description.

The UNESCO challenge, similarly to the OpenIDEO challenge, received ideas of various quality levels: ranging from excellent, good, mediocre, and finally to incomplete ones. To maintain comparability with our original results, but also taking into account that the focus of this work is using crowd filtering to sort the good ideas from the excellent ones, from the 176 ideas of the original dataset, we worked with the first 54 ideas in descending quality order (selecting 52 ideas, to match precisely the size of the OpenIdeo dataset was not possible, as the last four ideas of the UNESCO dataset had received the exact same number of comments, so we included them all). The number of clusters these ideas belonged to was 10, which again is comparable to the number of clusters of the original OpenIDEO dataset (8 clusters). The golden dataset consisted of the 16 most high-quality ideas, equal to the golden dataset size of the original experiment. Finally, similarly to the original experiment, we hired 50 Figure Eight crowd workers per experimental condition/algorithm.

Results, illustrated in Fig. 4.18, showed that the three compared algorithms performed similarly in relation one to the other as in the main experiment. In specific, the algorithms exhibited similar behavior in terms of performance (ROC curves, compare Fig. 4.18 and Fig. 4.14), page visit volume (compare Fig. 4.18 and Fig. 4.17) and median task time (compare Fig. 4.18 and Fig. 4.15). All three algorithms demonstrated a small drop in performance compared to the OpenIDEO experiment, and such a variation can be expected since the two datasets refer to different idea competitions and context. Observ-

ing this similarity in performance between the two datasets (OpenIDEO and UNESCO) reinforces the generalization potential of the proposed approach, for the specific stage of the innovation process (filtering the excellent from the good crowd ideas) and dataset size. The topic of generalization however is very broad. Different contexts, e.g. those involving ideas from experts (rather than from the crowd), or innovation stages (e.g. larger datasets from the initial stages of an idea contest that include a lot of incomplete or stub ideas) may affect generalization. Exploring these parameters can be the topic of further experiments and future research.

4.5.3.5 Key Assumptions

Below, we list the major assumptions of our work:

1. Our first assumption is that the subset of top ideas (golden set) which we use to benchmark algorithms truly reflects the best ideas. While we took utmost precautions to ensure the validity of this set, we believe that establishing a good golden set is necessary for comparing any ranking method.

2. We assume that workers can effectively perform the voting task within the time budget. It is possible that increasing or decreasing the time budget allocated to each worker may have an effect on the performance of different ranking algorithms.

3. We assume that the word embedding based clustering of ideas (or the categories provided for UNESCO's Youth Entrepreneurship ideation contest dataset) reflects how people categorize ideas. It maybe possible that people view diversity of ideas using some feature which we do not use.

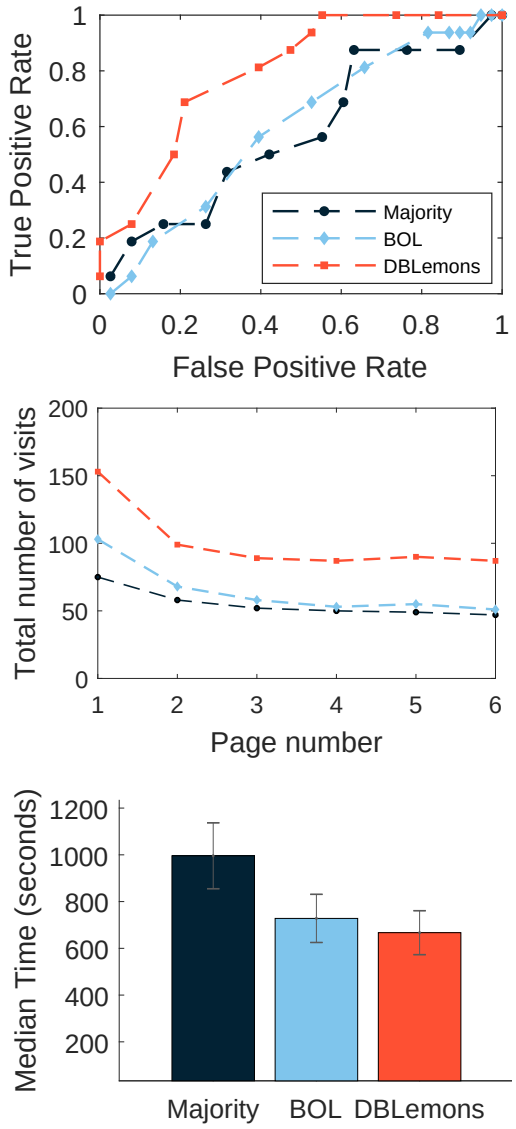

Figure 4.18: Examining the generalization of the proposed approach on a different dataset (UNESCO's Youth Entrepreneurship ideation contest). The three algorithms exhibit a similar behavior in comparison to one another as in the main experiment. a) Top left: ROC curve b) Top right: Page visit volume c) Median task time.

4. In our experiments, we do not show to a participant how many lemons or upvotes each idea has already received. This assumption ensures that people are less likely to be affected by preferential attachment.

5. We show ten ideas on each page, which implies that at a given time, a rater sees only a subset of ten ideas from the ranking. We assume that the number of ideas shown on each page does not significantly affect the performance of different ranking algorithms.

4.5.3.6 Limitations

In this work, we focused on the third (out of the four) stages of an OpenIDEO challenge. As such, our work can claim generalization only for the latter stages of open innovation, when the very low-quality or stub ideas have already been filtered out and it is harder to distinguish the good from the excellent ideas that will be retained for elaboration and funding. For a fully streamlined end-to-end crowd filtering process, which can return to the experts a minimum number of ideas and yet still contain almost all top-quality ones, in the future we aim to examine our approach on earlier stages of the open innovation process. Alternatively, it would be interesting to combine our existing approach with reference-based scoring models (e.g. [277]) that filter out large idea sets, by restricting crowd voter access to only a few representative ones. This would combine the advantages of both a quick filtering for the large mass of initial ideas, and of a more fine-grained one performed by our method for the latter stages. In the future, we would also like to examine this combination.

In the version of the algorithm used in this work, and for the DBLemons and BoL strategies, we gave voters a fixed budget of 10 lemons, corresponding to 19% of the idea corpus. This choice was made to be able to compare, to the best possible extent, our results with those of the original BoL strategy [154, 153]. However, changing the number of lemons is expected to affect (increase or decrease) the expressive power of voting. Consider for example the extreme case each user has only one lemon. It is likely that they will try to allocate this lemon to one of the worst ideas, but it is also likely that the ranking will be unable to distinguish between the rest of the ideas because of vote sparsity. On the other extreme, allocating a very large number of lemons will be time-consuming for the users, and it may affect the time of the task and possibly their accuracy. In the future we would like to systematically vary the number of lemons to study the effect of choice number on the filtering efficiency and on the time on the task. On a related remark, majority voting was implemented in this work as a single-voting strategy (where a user can allocate up to one vote per idea). This choice was motivated by the way majority voting is usually implemented in open innovation platforms (like OpenIdeo), and in order to be able to construct and compare with a realistic benchmark. In the future, it would also be interesting to examine a multi-voting adaptation of this strategy.

4.5.4 Concluding Remarks of Research Task 2

In this chapter, we demonstrated how diversity can be measured and optimized for diverse ranking and filtering applications. In the first research task, we proposed measuring diversity using submodular functions (discrete space) as well as Determinantal Point Processes (continuous space). We address key mathematical difficulties in using DPPs for ranking

and demonstrate the efficacy of our methods for online design communities. We showed how multi-objective optimization techniques can be deployed to find optimal ranking of ideas. In research task 2, we propose DBLemons, a new strategy for crowd-based idea filtering that combines the Bag of Lemons approach with a diversification of the idea concept space. Working with a dataset from an open innovation contest on women's safety, we show that DBLemons increases filtering efficiency and takes less time compared to majority voting (a popular open innovation filtering strategy), while compared to Bag of Lemons without diversity it also exhibits higher filtering efficiency. We attribute this to the larger number of idea comparisons made by voters in lesser time; this is possible since DBLemons shows representative ideas of all concepts early on. Overall, our proposed method contributes to improving trust in crowd-based idea filtering and hence it can help increase the strategic value of open innovation as an organizational governance choice.

4.6 Key Contributions

The main research contributions of this chapter are:

1. We define a new method for extending set-based diversity measures [15] to rank-based diversity measures. Our key insight lies in how to preserve a mathematical property called *sub-modularity* when computing diverse rankings; without it optimization becomes intractable.

2. We describe a method to balance high-quality versus diverse idea rankings through a quality and diversity trade-off front among rankings.

3. By evaluating two state-of-the-art approaches to compute diversity of item sets — sub-modular clustering and Determinantal Point Processes — we uncover the conditions under which one out-performs the other.

4. By using crowd experiments, we show that balancing idea quality and diversity improves the filtering of high-quality ideas. We argue that diverse ranking helps people make better comparative decisions, showing the benefit of incorporating diversity in information retrieval tasks.

5. We show that Bag of Lemons with dynamic ranking has similar filtering efficiency as Majority voting, but finds good ideas faster. This result confirms Klein and Garcia([154]) on the dynamic ranking setting.

4.7 Directions for Future Work

In this chapter, we defined multiple objectives for diverse ranking of a set of items and showed how multi-objective optimization methods can be leveraged to find diverse rankings for real-world applications. Next, we showed how diverse ranking benefited idea selection by combining it with crowd voting mechanism. These results have opened up directions of further inquiry for both theory and applications of diverse ranking.

- Learning appropriate DPP kernel or submodular function: How to learn a submodular function which estimates the ranking preference of people. In our first research task of this chapter, we compared two diversity functions — DPP and sum of concave functions. However, in doing so, we made two assumptions. First, we assumed that the choice of kernel for DPP (or the specific submodular function for

sum of concave functions) accurately represented how people viewed dissimilarity between items. However, it is possible that another kernel (or function) may better represent diversity of a set or ranking. Future work on learning such a function from human preference can lead to further improvements in this area. One possible direction of research can be learning a DPP kernel using people's prior preferences (*e.g.*basket-recommendation technique in [102]) and then using this kernel to rank order items.

- Learning vector representation of items: In many real-world applications, we do not know what vector representation should be used for items which are to be ranked. In task 1 of this chapter, we showed the applicability of our method to design sketches by finding vector representation using triplet embeddings derived from human responses. However, scaling ordinal comparisons to large idea sets is impractical due to the number of comparisons required from people. Future work can explore how new metric learning methods can be used to learn vector representations, with the goal to use these representations for diverse ranking.

- Eliciting user preference: In the second research task of this chapter, we showed a ranked list of all ideas to the participants. The participants vote on these ideas to find the best ones and the rank ordering changes after each participant votes. We extracted preferences using upvotes or lemons. However, the choice of preference elicitation may impact efficiency of filtering. It is unclear which preference elicitation methods lead to most efficient filtering in terms of time taken and top ideas found. Are pairwise comparisons better than upvotes for large datasets? These

questions point to the more fundamental question of how we use people as agents for searching optimal solution on a design space. The method of voting provides a structured approach on how these agents navigate the space.

- Efficient voting mechanism: In the second research task of this chapter, we compared multi-voting and majority voting for participants to filter ideas. However, when the collection of ideas becomes very large, many ideas may not be seen by the participants. Efficient filtering of a large collection of ideas is essential for success of open innovation model. To overcome this problem, future work can investigate methods on how to smartly elicit preferences from users. One possible direction to do this may be to show only a subset of ideas to each participant. Each participant may be prompted with comparisons that are dynamically chosen to ensure coverage over all items and high filtering efficiency. Diverse subset selection and methods outlined in [74] may be combined to explore this area of research. Another future research direction is to identify the minimum number of workers required for efficient DBLemons ranking, depending on idea complexity, cluster size and other variables.

- Interface Design for eliciting idea preferences: In open innovation contests, people are given access to the full set of candidate ideas [2]. Since this work has been about improving open innovation, we used the same type of interface, giving crowd voters the possibility to browse through all of the ideas if they wished to. However, there is also work [224] on eliciting preferences by prompting participants with comparisons that are dynamically chosen to ensure coverage and optimize for

speed. For instance, [73] studied usefulness of social choice functions in crowd-sourcing for participatory democracies and provided algorithms which efficiently elicit responses. In the future, it will be worth exploring the effect of these methods of preference elicitation for open innovation filtering.

- Impact of information cascades: In the second research task, voters did not see the ratings provided by others (although they are indirectly exposed to them due to the dynamic ranking applied on all three strategies). A future research direction could thus be to investigate the impact of information cascades on the open innovation crowd filtering problem.

- Different importance to different classes: The version of DBLemons used in this work gives equal weight to all clusters irrespective of their size. In the future it would be interesting to test alternative definitions of diversity, which give proportional weight to each cluster based on cardinality. We expect that this will mean that larger clusters will get more ideas at the top of the list.

- Learning diversity-utility trade-off: In this work we are interested in exploring the effect of diversity compared to the previously used methods of BoL and Majority voting. Therefore for the implementation of DBLemons we chose a λ value well above the cut-off limit of section 4.5.1.2, which gives the algorithm a clear preference for diversity. In the future it would be interesting to explore hybrid rankings, where a less clear preference on diversity is given to DBLemons, using λ values between zero and the cut-off limit. Below this limit, the lower the value the more the algorithm will resemble BoL and the higher the value the more the algorithm

will resemble DBLemons as it is described in this work. We note that the subject of how much one should diversify the ranking is a persistent question in the domain of recommender systems and also a domain- and context-specific problem.

4.8 Conclusion of Chapter 4

In this chapter, we first propose a method to measure diversity of sets and ranked lists of items. These measures were combined with quality to simultaneously maximize the quality and diversity of a ranking. Specifically, this research added the following new pieces of knowledge: 1) how to extend set-based diversity metrics to rank-based diversity measures, 2) how to rank ideas by diversity in polynomial time using a greedy strategy with theoretical performance guarantees, 3) how to trade-off quality and diversity when ranking ideas, and 4) how one can use the determinant of a design space to uncover properties of that space (such as how much quality one has to sacrifice to gain diversity) and the extent to which one can achieve compression in the ideas one considers (via comparisons along the quality-to-diversity trade-off front).

We demonstrated and validated the above contributions using both benchmark datasets and 606 real-world design ideas from an OpenIDEO challenge. We showed that our method produces higher quality, more diverse rankings than competing techniques. Our findings have several implications both for ranking items and studying ideation at large scale. First, Fig. 4.2 showed that, out of 606 ideas, only 36 unique solutions appeared across any portion of the trade-off front in top 10 ideas, from high-quality to high-diversity. This implies that, even without picking a location on the trade-off front, we

can achieve substantial compression in the "minimal set" of inspiring ideas a designer might consider — roughly 6% in our example. In the real-world scenario we analyzed, this meant reducing designer effort from roughly 25 hours to 90 minutes. Second, when trading off diversity and quality, we found that maximizing diversity without considering quality produced less useful ideas than considering the combination. This implies that we need better automated quality metrics for ideas — similar to those researchers have proposed for diversity or variety — if we hope to scale up our ability to evaluate or inspire creative ideas.

In the second part of this chapter, we propose DBLemons, a new strategy for crowd-based idea filtering that combines the Bag of Lemons approach with diverse ranking. Working with a dataset from an open innovation contest on women's safety, we show that DBLemons increases filtering efficiency and takes less time compared to majority voting (a popular open innovation filtering strategy), while compared to Bag of Lemons without diversity it also exhibits higher filtering efficiency. We attribute this to the larger number of idea comparisons made by voters in lesser time; this is possible since DBLemons shows representative ideas of all concepts early on. Overall, our proposed method contributes to improving trust in crowd-based idea filtering and hence it can help increase the strategic value of open innovation as an organizational governance choice.

Title	Set (Most Diverse (A), Highest Quality (C) and Radial Set (B))
Building 'Transparency' App (updated)	C
Eatcyclopedia: A Phone App to Help Connect and Inform	C
Hold Seasonal "Open House" Days at Local Farms	C
The Farmer and The Chef	C, B
Closing the Farmers Market Loop	C, B
Market Days + Food Trucks = Serving Low-income Neighborhoods	C, B
Redesign the supermarket layout based on food miles... UPDATED	C, B
Window to the Farm	C, B
Public Kitchen	C, B
A celebration of imperfection	C, B
50 Within 50	B
Traveling Movie Theater on Farms	B
Fruit Trees instead of Fences	B, A
Branded Clothing	A
Intensive two-week Internship on farms: Interns will teach others when they come back to the city	A
Trick yourself into sustainable buying	A
Trade & resell network for CSA share-holders. Specific to central pick-up location for many CSA programs.	A
Dentell	A
Install Greenhouses at Train Stations	A
A new youth movement: Healthy Eating and living	A
fruity roofs	A
Hack Cooking to Make it Appealing	A

Table 4.3: OpenIDEO ideas on the trade-off front.

Idea 1	Idea 2
This idea proposes a participatory planning process, which enables communities to design safer public spaces for women through safety audits. Safety auditing is the process of gathering data about the safety of a place, and it is usually based on data gathered through smartphones. Unfortunately smartphone usage is limited in countries like India. To solve this issue, local community members and volunteers conduct safety audits of public spaces and data is collected through not only mobile, but also through online or face-to-face meetings. The analyzed data is displayed on a large interactive map in an open public space such as a park. This map can then be used as a canvas on which local community members can write and draw their reactions and suggestions to the safety audit information provided. The idea will be evaluated by tracking change in safety parameters over time.	This idea proposes a platform that enables women to work from home. The platform consists of a network of income generation modules, providing women with appropriate equipment to work, especially in poor housing areas. These physical modules can be easily attached to low-income houses, with properly designed working spaces and online connectivity. Women can choose their preferred module based on local skills and demands. The platform will help reduce the time of traveling to work and directly improve women's safety, while also giving them more time for childcare. Continually educating women will also help them understand their work rights and the benefits of not including children in work. The idea will be implemented gradually, from system design and mapping the viable income generation alternatives, to prototyping and launching the platform. The evaluation will be made through qualitative and quantitative data that assesses the platform's impact on empowering women.

Table 4.4: Two conceptually different ideas incorrectly clustered together by automated clustering. Manual clustering assigned them to different clusters: the first idea to "Public Spaces" and the second idea to "Employment".

Chapter 5: Conclusion

5.1 Motivation

The overarching goal of this dissertation is to enable design democratization, which means to help distributed teams of people from around the world collaborate to design products. Manufacturing employment in the US has decreased by almost thirty percent in the last twenty years [1]. People who were traditionally employed in the manufacturing sector are increasingly looking for new employment opportunities. Many of these people have vast experience in their specific area of work but have limited mobility to move to places with newer job opportunities. This presents us with challenges to create new employment opportunities for these people. One possible solution is to enable these people to design custom products from the comfort of their homes. Online design contests is a step in that direction.

An example of the promise of design contests is a guy named Edgar Sarmiento from Colombia, who had little job opportunities in his hometown. He designed Olli — a self-driving bus, by participating remotely in an online design contest organized by Local Motors. These design contests often receive thousands of ideas, which are difficult to process in a short amount of time. We believe machine learning and computing methods

[1] Bureau of Labor Statistics `https://data.bls.gov/pdq/SurveyOutputServlet`

can help improve online design contests and enable design teams to work remotely. By developing ways to filter a large collection of ideas, measure their creativity and form teams to both generate and evaluate ideas, this dissertation provides tools to enable design democratization. We addressed these problems by asking ourselves three questions: 1. How does one reliably measure the creativity of ideas? 2. How does one form teams to evaluate design ideas? 3. How does one filter good ideas out of hundreds of submissions?

Chapter 2, 3 and 4 provide computational methods to address these questions. The underlying theme in all these chapters was ways to measure, learn and optimize novelty and diversity of a set of items. By doing so, our primary research question, "How (and why) does one measure, learn and optimize novelty and diversity of a set of items?" ties together the findings of each of these chapters.

5.2 Dissertation Summary

The fundamental scientific contributions of this dissertation are in the measurement, learning, and optimization of novelty and diversity. The methods developed here can help us design new systems. In Chapter 2, we identified issues of validity, explainability, and repeatability with two design metrics — novelty and variety, and proposed computational metrics to address those issues. Reliably measuring the creativity of ideas can help design managers in using those metrics to process large collections of ideas. By uncovering factors important for creativity estimation, our methods can be used for training raters and designers to focus on those factors. In Chapter 3, we presented quantitative approaches to balancing diversity and efficiency for a bipartite matching problem. These algorithms can

be used to match people to their peers such that the team is both diverse and high quality. In Chapter 4, we proposed methods to filter good ideas from a large collection using a combination of multi-objective optimization and submodular functions. These methods are key to improve the efficiency of idea filtering.

5.3 Discussion

In this work, we focused on diversity metric with pre-defined structure. In chapter two, we introduced HHID, which is a supermodular diversity metric based on a quadratic function. By introducing weight parameters for different functional levels, we provided the flexibility to adapt the metric to one's domain. Similar quadratic based diversity function was used in forming diverse teams in Research Task 1 and Research Task 2 of the third chapter. The matching optimization problem was more complex to solve, due to additional constraints on how many people are needed in each team and the maximum number of tasks a person was willing to do. In different chapters, we showed how to optimize these functions and the effect of optimizing these functions on different domains. In the third chapter, we introduced a square-root based submodular objective function for online diverse team formation. A similar square-root based submodular objective function was used to measure diversity in both research tasks of the fourth chapter. While the goal of the third chapter was to form diverse teams with sequential arrival of people one at a time, the diversity function introduced in next chapter was used to rank-order a set of ideas such that the ranking balanced high quality and idea diversity.

All the above objective functions required that the set of items are clustered into

groups. For matching of reviewers to papers, these groups were expertise area of the reviewer. Each paper expected to receive a team of reviewers with different expertise. For matching people to teams, these groups were gender, educational qualification, and other demographic information. Each task expected to receive people with different genders. For measuring variety of design ideas, these groups were design principles which were used by the ideas. For movies, the groups were genres of movies. The basic idea was that diversity was defined in all these chapters as coverage over a fixed number of groups using a submodular or a supermodular function. The key takeaway message was to define diversity functions which are computationally tractable and then propose combinatorial optimization algorithms which cater to different complexities of the problem definition.

However, the above formulations were defined only for discrete domains. It is possible that domains do not have well defined groups into which items are categorized. To address these problems, we proposed Determinantal Point Processes. To define a DPP, one requires a semi-positive definite kernel matrix. While DPPs' were traditionally used to measure diversity of a set, we extended DPP formulation to the rank-ordering of ideas in the fourth chapter. We also compared DPPs' to the square-root diversity function and showed that DPPs' are more useful when one does not have an estimate of the number of categories in their dataset. If one has an estimate of the number of categories, then we show that they can cluster their data into discrete categories before applying a square root or a quadratic function. The key takeaway message was to develop methods to measure diversity in continuous domains too and demonstrating the promise of DPPs' as a vehicle to do so.

To apply DPPs successfully to different domains, one needs to define an appropriate

kernel function. Such a kernel function is generally derived from vector representations of items. Similarly, if items are to be clustered into groups, the input to a clustering algorithm is often the vector representation of all ideas. Using the right representation scheme is fundamental to all our research questions. To represent ideas in the fourth chapter, we use TF-IDF vectors for text representation. In matching applications, we represent each reviewer by a vector of their research interests and each movie by a vector of the genres that movie contains. For design sketches in the second research task of the second chapter, we use three hierarchical attributes which were hand-coded. While deciding what representation is appropriate using past-experience is commonly done, learning the representation scheme from people reduces possible biases. We addressed this question directly in the first chapter to find design embeddings for sketches. We first learned the embeddings from people by asking them simple questions. These embeddings were then used to define novelty functions. Learning representation schemes for items with complex relationships between them is an interesting area of future research. The key takeaway message was to focus on learning representation schemes to better understand how items are defined as unique or novel.

5.4 Key Limitations

The limitations of the research methods presented in this dissertation are listed in each chapter. However, below we discuss a few limitations which were common across the thesis.

- We need scalable algorithms and large datasets: Due to constraints of cost, time

and practicality, many of our experiments were conducted on small to medium scale datasets. We established datasets to compare diversity metrics in the second chapter. We also provided embeddings for ten design sketches. However, to learn complex patterns and relationships from design items, we need much larger datasets. Establishing these datasets is non-trivial and without large datasets, developing better models is restricted. A similar problem occurred in the Computer Vision community, before the ImageNet dataset [223] was released. However, ImageNet required tagging and annotation of object categories by crowd-workers, a task which did not require specialized training. In Design applications, establishing large scale datasets of designs require experts, who are more difficult to recruit and more expensive. The trade-off is often of time and cost. It can be costly to establish large datasets, so one has to ensure that most items add sufficient value (if all ideas are slight variants of each other, the cost of getting expert ratings on all of them is less useful than of all ideas are different from each other). Future work could address how to perform a collection of large datasets in an optimal fashion (*e.g.*, using Active Learning) and bound the number of comparisons one would need to collect.

The problem of scalability occurs not only in establishing datasets but also in developing better algorithms. In our first matching algorithm using MIQP, for a medium-sized graph of 1000 total nodes, the number of total terms in the objective function was of the order of ten billion. This led us to develop auxiliary graph approaches, which is an order of magnitude faster. However, the algorithms are still not scalable to be applied to graphs of millions of nodes. Solving this scalability issue (under

reasonable approximations) for diversity-related problems presents an interesting challenge, which future researchers can investigate.

- Learning from people is challenging: To train supervised machine learning algorithms, we often need from people. In the second chapter, we trained models using triplet responses. We found that people are poor at making their idea maps directly on a pin-board but do relatively well when they answer simple triplet queries. However, there are myriad different ways to collect responses from people. Instead of a triplet, one can show four or five items and ask the participant to find the most similar item. One can ask the participants to cluster a set of items into different groups and derive triplet responses from such grouping. One can directly ask a similarity rating on a Likert scale. This brings us to the question, how does one decide which method of response elicitation is better and under what conditions should one method be chosen over another? In this research, we selected our methods of response collection based on past literature in other related domains. However, we did not conduct an exhaustive study of which preference elicitation methods are more suitable for each application. Learning from people also ties to the scalability issue discussed before. Due to limited attention span and exhaustion, one is limited in the number of responses they can get from each individual. People's choices may change with time both due to exhaustion or due to exposure to the task at hand. These temporal effects may limit the applicability of our findings (like design embeddings in Chapter 2 and Bag of Lemons in Chapter 4), which consider the entire set of responses together.

- Effect of modeling choices: Every algorithm and method comes with its own set of assumptions and limitations. For instance, we showed in Chapter 2 that the choice of using SVS [235] as a variety metric implies that it is not sensitive to a large percentage of designs. Similarly, in using HHID metric with quadratic exponent, we make a choice. If we let the exponent go to infinity, it means the variety metric will predominantly be affected by the number of items in the largest group. If we decrease the power to zero, the metric collapses to counting the number of unique groups in the set. Hence, by choosing quadratic power in the metric, we make a modeling choice. Although this choice was necessitated by the need of getting efficient optimization methods (as we saw in Chapter 4 using a MIQP), the choice restricts the type of variety or diversity we can measure. By minimizing the number of choices we make in the methods, we tend to increase the generality of the models and reduce the number of assumptions needed. Future research can investigate ways to learn metrics directly from data and study how different modeling choices affect real-world deployment.

5.5 Summary of Future Research Directions

This dissertation has led us to many new open questions which can be addressed in future research. The motivation behind these open questions is to identify how best we can achieve design democratization, where teams of globally distributed people work together to create physical products. To achieve design democratization, it is important to identify current gaps in both theory and practice, which need to be solved to enable people to

participate in the design process from remote locations. While we presented many open questions in each of the three chapters, when one looks at the problem holistically, a few key themes emerge.

5.5.1 Creativity Estimation

The first theme is inspired by the question, "Can computers estimate the creativity of the next generation autonomous car?" Estimating the creativity of something we have not seen before is difficult for humans too, and future research can focus on how we can build machine learning-enabled creativity estimation tools. The output of such a system should be an accurate estimation of the creativity of new items based on learnings from the past. For computers to reliably estimate creativity, they have to learn to estimate both the quality and novelty of an item. Quality estimation may require complex simulation models or domain-specific knowledge. For novelty, understanding what factors are used by experts and novices in their valuation of items and judging their similarity is key. However, learning the similarity between items is non-trivial due to a large number of queries needed to explore all options. Future research should focus on better methods to capture how humans assess the similarity between items. Learning similarity is especially hard when an item appears which has not been seen before. By developing computer-aided raters for large collections of unseen ideas, we can improve the innovation potential of our society and develop more efficient ways for products to reach the market.

5.5.2 Team Formation

The second theme is inspired by the question, "How can we develop practical methods for team formation?" For practical team formation, many factors need to be taken into account like team goals, budget, task-type, team interactions, diversity *etc.* An open challenge is to understand how much team diversity is needed for a given project and based on what attributes? In the algorithms we developed in this dissertation, we made two assumptions: the end-user knows how much diversity they want in their team for a given project and they also know what factors are key in deciding whether a team is diverse or not (*i.e.* the end-user knows whether they want their team to have diverse skillsets, diverse gender distribution or people from multiple countries.) However, practically many managers may not know what type of diversity they want in their teams. To enable team formation, it will be important to understand what factors are truly important in achieving the goals of a team (say performance on a task).

5.5.3 Learning Representation

The third theme is learning better representation schemes for complex ideas. Representing ideas which are in heterogeneous format is difficult and is often needed for tasks like estimating the similarity between ideas, training recommendation engines *etc.* Without using meaningful representation, most algorithms for idea filtering will not give desirable results. However, we do not have a common language to represent different types of items. For instance, in this dissertation, we used triplet embeddings for sketches and topic models for word documents. What happens when a design item has a CAD file

showing the 3-D assembly, a text document describing its usage and a photo rendering showing how it looks? To process a large collection of such unstructured data, developing a common language for representing complex heterogeneous items is needed.

5.5.4 Unification of Metrics

The fourth theme is the possible relationship between diversity and novelty. Both novelty and diversity are derived from similarities between items and we contend that they are inter-related. The most novel idea in a set is the one which is most different from everyone else. If we remove this most novel idea from a set of existing ideas, then the remaining set should become the least diverse set of a given size. Hence, we believe that it should be possible to measure the novelty of an idea by taking the difference in diversity scores. Future research should investigate this relationship, to develop a unified model of design metrics using common mathematical principles.

5.5.5 Learning Submodular Functions

Finally, the last theme is around learning submodular functions from human responses. In this dissertation, we pre-defined the submodular function (like the sum of concave functions) and used it for defining objective functions defining coverage over items. Similarly, there has been much interest in machine learning and algorithmic game theory communities on understanding and using submodular functions to model diversity or coverage. Despite this substantial interest, little is known about their learnability from data. While [32] has provided a few theoretical algorithms to learn submodular functions, the methods still require many queries to be practically used in learning submodular functions from

people. We hope this dissertation provides useful building blocks for these directions of inquiry and leads to new directions of research.

Bibliography

[1] I. merriam-webster, merriam-webster — an encyclopedia britannica company. https://www.merriam-webster.com/dictionary/novelty. Accessed: 2018-10-01.

[2] Simon à Campo, Vasssilis-Javed Khan, Konstantinos Papangelis, and Panos Markopoulos. Community heuristics for user interface evaluation of crowdsourcing platforms. *Future Generation Computer Systems*, 2018.

[3] Selcuk Acar and Mark A Runco. Latency predicts category switch in divergent thinking. *Psychology of Aesthetics, Creativity, and the Arts*, 11(1):43, 2017.

[4] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 24(5):896–911, 2012.

[5] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18, 2007.

[6] Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1405–1424. SIAM, 2014.

[7] Shipra Agrawal, Morteza Zadimoghaddam, and Vahab Mirrokni. Proportional allocation: Simple, distributed, and diverse matching with high entropy. In *International Conference on Machine Learning (ICML)*, pages 99–108, 2018.

[8] Saba Ahmadi, Faez Ahmed, John P Dickerson, Mark Fuge, and Khuller Samir. On diverse bipartite b-matchings. In *Under submission*, 2019.

[9] Faez Ahmed, John P Dickerson, and Mark Fuge. Diverse weighted bipartite b-matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 35–41. AAAI Press, 2017.

[10] Faez Ahmed, John P. Dickerson, and Mark Fuge. Diverse weighted bipartite b-matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 35–41. AAAI Press, 2017.

[11] Faez Ahmed, John P. Dickerson, and Mark Fuge. Diverse weighted bipartite b-matching. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 35–41, 2017.

[12] Faez Ahmed, John P Dickerson, and Mark Fuge. Online diverse team formation. In *Under submission*, 2019.

[13] Faez Ahmed and Mark Fuge. Capturing winning ideas in online design communities. In *20th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, Portland, USA, February 2017. ACM.

[14] Faez Ahmed and Mark Fuge. Ranking ideas for diversity and quality. *Journal of Mechanical Design*, 140(1):011101, 2018.

[15] Faez Ahmed, Mark Fuge, and Lev D. Gorbunov. Discovering diverse, high quality design ideas from a large corpus. In *ASME International Design Engineering Technical Conferences*, Charlotte, USA, August 2016. ASME.

[16] Faez Ahmed, Mark Fuge, Sam Hunter, and Scarlett Miller. Unpacking subjective creativity ratings: Using embeddings to explain and measure idea novelty. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T06A003–V007T06A003. American Society of Mechanical Engineers, 2018.

[17] Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Interpreting idea maps: Pairwise comparisons reveal what makes ideas novel. volume 141, page 021102. American Society of Mechanical Engineers, 2019.

[18] Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Interpreting idea maps: Pairwise comparisons reveal what makes ideas novel. *Journal of Mechanical Design*, 141(2):021102, 2019.

[19] Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Measuring and optimizing design variety using herfindahl index. In *ASME International Design Engineering Technical Conferences*, Anaheim, USA, August 2019. ASME.

[20] Kamal Ali and Wijnand Van Stam. Tivo: making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 394–401. ACM, 2004.

[21] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81, 2013.

[22] Maha Alsayasneh, Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Mae Borromeo, Motomichi Toyama, and Jean-Michel Renders. Personalized and diverse task composition in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 30(1):128–141, 2018.

[23] Teresa M Amabile. *Creativity in context: Update to the social psychology of creativity*. Hachette UK, 1996.

[24] Teresa M Amabile and Julianna Pillemer. Perspectives on the social psychology of creativity. *The Journal of Creative Behavior*, 46(1):3–15, 2012.

[25] Ehsan Amid and Antti Ukkonen. Multiview triplet embedding: Learning attributes in multiple maps. In *International Conference on Machine Learning*, pages 1472–1480, 2015.

[26] Ehsan Amid, Nikos Vlassis, and Manfred K Warmuth. Low-dimensional data embedding via robust ranking. *arXiv preprint arXiv:1611.09957*, 2016.

[27] Mariia Anapolska, Christina Büsing, and Martin Comis. Minimum color-degree perfect b-matchings. In *Working paper. Early version appeared in the 16th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, page 13, 2018.

[28] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Inigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

[29] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1742–1748, 2015.

[30] Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1497–1514, 2014.

[31] John Baer. Domain specificity and the limits of creativity theory. *The Journal of Creative Behavior*, 46(1):16–29, 2012.

[32] Maria-Florina Balcan and Nicholas JA Harvey. Submodular functions: Learnability, structure, and optimization. *SIAM Journal on Computing*, 47(3):703–754, 2018.

[33] Murray R Barrick, Greg L Stewart, Mitchell J Neubert, and Michael K Mount. Relating member ability and personality to work-team processes and team effectiveness. *Journal of Applied Psychology*, 83(3):377, 1998.

[34] W Beitz and G Pahl. Engineering design: a systematic approach. *MRS BULLETIN*, 71, 1996.

[35] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[36] Nawal Benabbou, Mithun Chakraborty, Xuan-Vinh Ho, Jakub Sliwinski, and Yair Zick. Diversity constraints in public housing allocation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 973–981. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[37] Ivo Blohm, Christoph Riedl, Johann Füller, and Jan Marco Leimeister. Rate or trade? identifying winning ideas in open idea sourcing. *Information Systems Research*, 27(1):27–48, 2016.

[38] Rubi Boim, Tova Milo, and Slava Novgorodov. Diversification and refinement in collaborative filtering recommender. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 739–744. ACM, 2011.

[39] Alexei Borodin. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.

[40] P. Allen Bradley. MovieLens collaborative filtering. `https://github.com/bradleypallen/keras-movielens-cf`, 2016.

[41] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[42] David C Brown. Problems with the calculation of novelty metrics. In *Proc. Design Creativity Workshop, 6th Int. Conf. on Design Computing and Cognition (DCC'14)*, 2014.

[43] D Bryne, GL Clore Jr, and P Worchel. The effect of economic similarity-dissimilarity as determinants of attraction. *Journal of Personality and Social Psychology*, 4:220–224, 1966.

[44] David V Budescu and Eva Chen. Identifying expertise to extract the wisdom of crowds. *Management Science*, 61(2):267–280, 2014.

[45] Alex Burnap, Yanxin Pan, Ye Liu, Yi Ren, Honglak Lee, Richard Gonzalez, and Panos Y Papalambros. Improving design preference prediction accuracy using feature learning. *Journal of Mechanical Design*, 138(7):071404, 2016.

[46] Alex Burnap, Yi Ren, Richard Gerth, Giannis Papazoglou, Richard Gonzalez, and Panos Y Papalambros. When crowdsourcing fails: A study of expertise on crowdsourced design evaluation. *Journal of Mechanical Design*, 137(3):031101, 2015.

[47] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

[48] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[49] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. ACM, 1998.

[50] Ben Carterette. An analysis of np-completeness in novelty and diversity ranking. In *Conference on the Theory of Information Retrieval*, pages 200–211. Springer, 2009.

[51] HERNAN Casakin and SHULAMIT Kreitler. The nature of creativity in design. *Studying Designers*, 5:87–100, 2005.

[52] Pablo Castells, Neil J Hurley, and Saul Vargas. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*, pages 881–918. Springer US, 2015.

[53] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation. *ArXiv e-prints*, July 2017.

[54] Amaresh Chakrabarti, Kristina Shea, Robert Stone, Jonathan Cagan, Matthew Campbell, Noe Vargas Hernandez, and Kristin L Wood. Computer-based design synthesis research: an overview. *Journal of Computing and Information Science in Engineering*, 11(2):021003, 2011.

[55] Joel Chan, Steven Dang, and Steven P Dow. Comparing different sensemaking approaches for large-scale ideation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2717–2728. ACM, 2016.

[56] Joel Chan, Katherine Fu, Christian Schunn, Jonathan Cagan, Kristin Wood, and Kenneth Kotovsky. On the benefits and pitfalls of analogies for innovative design: Ideation performance based on analogical distance, commonness, and modality of examples. *Journal of mechanical design*, 133(8):081004, 2011.

[57] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592, 2011.

[58] Laurent Charlin and Richard Zemel. The toronto paper matching system: an automated paper-reviewer assignment system. 2013.

[59] Laurent Charlin and Richard S Zemel. The Toronto paper matching system: an automated paper-reviewer assignment system. In *International Conference on Machine Learning (ICML)*, 2013.

[60] Cheng Chen, Lan Zheng, Venkatesh Srinivasan, Alex Thomo, Kui Wu, and Anthony Sukow. Conflict-aware weighted bipartite b-matching and its application to e-commerce. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1475–1488, 2016.

[61] Liang Chen and De Liu. *Comparing strategies for winning expert-rated and crowd-rated crowdsourcing contests: First findings*, volume 1, pages 97–107. 12 2012.

[62] Liang Chen, Pei Xu, and De Liu. Experts versus the crowd: a comparison of selection mechanisms in crowdsourcing contests. 2016.

[63] Wei Chen, Jonah Chazan, and Mark Fuge. How designs differ: Non-linear embeddings illuminate intrinsic design complexity. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V02AT03A014–V02AT03A014. American Society of Mechanical Engineers, 2016.

[64] I Chiu and LH Shu. Investigating effects of oppositely related semantic stimuli on design concept creativity. *Journal of Engineering Design*, 23(4):271–296, 2012.

[65] Po-Wen Chiu and CL Bloebaum. Hyper-Radial Visualization (HRV) with weighted preferences for multi-objective decision making. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 10–12, 2008.

[66] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[67] David Clutterbuck. *Coaching the team at work*. Nicholas Brealey Publishing, 2011.

[68] Clyde H Coombs and George S Avrunin. Single-peaked functions and the theory of preference. *Psychological review*, 84(2):216, 1977.

[69] Taylor H Cox and Stacy Blake. Managing cultural diversity: Implications for organizational competitiveness. *The Executive*, pages 45–56, 1991.

[70] Arthur J Cropley. Defining and measuring creativity: Are creativity tests worth using? *Roeper review*, 23(2):72–79, 2000.

[71] Nigel Cross and Robin Roy. *Engineering design methods*, volume 4. Wiley New York, 1989.

[72] Thomas Cummings. *Organization development and change*. Wiley Online Library, 2009.

[73] Tanja Aitamurto Helene Landemore David Timothy Lee, Ashish Goel. Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[74] Luca De Alfaro, Vassilis Polychronopoulos, and Neoklis Polyzotis. Efficient techniques for crowdsourced top-k lists. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[75] Douglas L Dean, Jillian M Hender, Thomas L Rodgers, and Eric L Santanen. Identifying Quality, Novel, and Creative Ideas: Constructs and Scales for Idea Evaluation. *Journal of the Association for Information Systems*, 7(10):646–698, 2006.

[76] Kalyanmoy Deb and Shivam Gupta. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, 43(11):1175–1204, 2011.

[77] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[78] Çağatay Demiralp, Michael S Bernstein, and Jeffrey Heer. Learning perceptual kernels for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12):1933–1942, 2014.

[79] Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 137–144. ACM, 2012.

[80] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. An analysis of users' propensity toward diversity in recommendations. In *ACM Conference on Recommender Systems (RecSys)*, pages 285–288, 2014.

[81] John P. Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Balancing relevance and diversity in online bipartite matching via submodularity. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[82] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. Demographics and dynamics of mechanical turk workers. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 135–143, New York, NY, USA, 2018. ACM.

[83] Andy Dong. The latent semantic approach to studying design team communication. *Design Studies*, 26(5):445–461, 2005.

[84] Wei Dong, Kate Ehrlich, Michael M Macy, and Michael Muller. Embracing cultural diversity: Online social ties in distributed workgroups. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 274–287. ACM, 2016.

[85] Kees Dorst and Nigel Cross. Creativity in the design process: co-evolution of problem–solution. *Design studies*, 22(5):425–437, 2001.

[86] Steven Dow, Julie Fortuna, Dan Schwartz, Beth Altringer, Daniel Schwartz, and Scott Klemmer. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2807–2816. Acm, 2011.

[87] Shristi Drolia, Shrey Rupani, Pooja Agarwal, and Abheejeet Singh. Automated essay rater using natural language processing. *International Journal of Computer Applications*, 163(10):44–46, Apr 2017.

[88] Joanna Drummond, Andrew Perrault, and Fahiem Bacchus. Sat is an effective and complete method for solving stable matching problems with couples. In *IJCAI*, pages 518–525, 2015.

[89] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[90] N Dylla. Thinking methods and procedures in mechanical design. *Mechanical design, technical university of Munich, PhD*, 1991.

[91] Nadine Escoffier and Bill McKelvey. The wisdom of crowds in the movie industry: Towards new solutions to reduce uncertainties. *International Journal of Arts Management*, 17(2):52, 2015.

[92] Ali Farhang-Mehr and Shapour Azarm. An information-theoretic entropy metric for assessing multi-objective optimization solution set quality. *Journal of Mechanical Design*, 125(4):655–663, 2003.

[93] Azarm Farhang-Mehr and Shapour Azarm. Entropy-based multi-objective genetic algorithm for design optimization. *Structural and Multidisciplinary Optimization*, 24(5):351–361, 2002.

[94] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

[95] David Fisher, Ashish Jain, Mostafa Keikha, WB Croft, and Nedim Lipka. Evaluating ranking diversity and summarization in microblogs using hashtags. Technical report, Technical report, University of Massachusetts, 2015.

[96] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–765, 2006.

[97] Mark Fuge and Alice Agogino. How online design communities evolve over time: the birth and growth of OpenIDEO. In *ASME International Design Engineering Technical Conferences*. ASME, August 2014.

[98] Mark Fuge and Alice Agogino. How online design communities evolve over time: the birth and growth of OpenIDEO. In *ASME International Design Engineering Technical Conferences*, Buffalo, U.S.A., August 2014.

[99] Mark Fuge, Josh Stroud, and Alice Agogino. Automatically inferring metrics for design creativity. *ASME Paper No. DETC2013-12620*, 2013.

[100] Mark Fuge, Kevin Tee, Alice Agogino, and Nathan Maton. Analysis of collaborative design networks: A case study of openideo. *Journal of Computing and Information Science in Engineering*, 14(2):021009, 2014.

[101] Mark Fuge, Kevin Tee, Alice Agogino, and Nathan Maton. Analysis of collaborative design networks: A case study of OpenIDEO. *Journal of Computing and Information Science in Engineering*, 14(2):021009+, March 2014.

[102] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Low-rank factorization of determinantal point processes. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[103] Corrado Gini. Variabilità e mutabilità. *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi*, 1912.

[104] Andrew V. Goldberg and Tomasz Radzik. A heuristic improvement of the bellman-ford algorithm, 1993.

[105] Paul Gölz and Ariel D Procaccia. Migration as submodular optimization. *arXiv preprint arXiv:1809.02673*, 2018.

[106] Thomas Görzen and Dennis Kundisch. Can the crowd substitute experts in evaluating creative jobs? the case of business models. In *ECIS*, pages Research–in, 2016.

[107] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[108] Matthew Green, Carolyn Conner Seepersad, and Katja Hölttä-Otto. Crowdsourcing the evaluation of creativity in conceptual design: A pilot study. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T07A016–V007T07A016. American Society of Mechanical Engineers, 2014.

[109] Joseph H Greenberg. The measurement of linguistic diversity. *Language*, 32(1):109–115, 1956.

[110] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[111] Lidia Gryszkiewicz, Ioanna Lykourentzou, and Tuukka Toivonen. Innovation labs : leveraging openness for radical innovation? *Journal of Innovation Management*, 4(4):68–97, 2016.

[112] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016.

[113] J Richard Hackman and Greg R Oldham. Motivation through the design of work: Test of a theory. *Organizational behavior and human performance*, 16(2):250–279, 1976.

[114] Siavash Haghiri, Debarghya Ghoshdastidar, and Ulrike von Luxburg. Comparison-based nearest neighbor search. In *Artificial Intelligence and Statistics*, pages 851–859, 2017.

[115] Wafa Hammedi, Allard CR van Riel, and Zuzana Sasovova. Antecedents and consequences of reflexivity in new product idea screening. *Journal of Product Innovation Management*, 28(5):662–679, 2011.

[116] Stevan Harnad. Validating research performance metrics against peer rankings. *Ethics in science and environmental politics*, 8(1):103–107, 2008.

[117] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[118] David A Harrison and Katherine J Klein. What's the difference? diversity constructs as separation, variety, or disparity in organizations. *Academy of Management Review*, 32(4):1199–1228, 2007.

[119] David A Harrison, Kenneth H Price, Joanne H Gavin, and Anna T Florey. Time, teams, and task performance: Changing effects of surface-and deep-level diversity on group functioning. *Academy of Management Journal*, 45(5):1029–1045, 2002.

[120] Jingrui He, Hanghang Tong, Qiaozhu Mei, and Boleslaw Szymanski. Gender: A generic diversified ranking algorithm. In *Advances in Neural Information Processing Systems*, pages 1142–1150, 2012.

[121] Daniel Henderson, Kevin Helm, Kathryn Jablokow, Seda McKilligan, Shanna Daly, and Eli Silk. A comparison of variety metrics in engineering design. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T06A004–V007T06A004. American Society of Mechanical Engineers, 2017.

[122] Beth A Hennessey and Teresa M Amabile. Consensual assessment. *Encyclopedia of creativity*, 1:347–359, 1999.

[123] Joel West Henry Chesbrough, Wim Vanhaverbeke. *Open Innovation: Researching a New Paradigm*. Oxford University Press, 2006.

[124] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377. Association for Computational Linguistics, 2016.

[125] Albert O Hirschman. The paternity of an index. *The American economic review*, 54(5):761–762, 1964.

[126] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[127] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems (TOIS)*, 31(4):17, 2013.

[128] Lawrence Holpp. *Managing teams*. McGraw Hill Professional, 1999.

[129] Tom Hope, Joel Chan, Aniket Kittur, and Dafna Shahaf. Accelerating innovation through analogy mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 235–243, New York, NY, USA, 2017. ACM.

[130] John Joseph Horton. The effects of algorithmic labor market recommendations: evidence from a field experiment, 2017. To appear, Journal of Labor Economics.

[131] Sujin K Horwitz and Irwin B Horwitz. The effects of team diversity on team outcomes: A meta-analytic review of team demography. *Journal of Management*, 33(6):987–1015, 2007.

[132] Chu-Yi Huang, Yen-Shen Chen, Youn-Long Lin, and Yu-Chin Hsu. Data path allocation based on bipartite weighted matching. In *ACM/IEEE Design Automation Conference*, 1991.

[133] Eelko K.R.E. Huizingh. Open innovation: State of the art and future perspectives. *Technovation*, 31(1):2 – 9, 2011. Open Innovation - ISPIM Selected Papers.

[134] Stephen E Humphrey, Frederick P Morgeson, and Michael J Mannor. Developing a theory of the strategic core of teams: a role composition model of team performance. *Journal of Applied Psychology*, 94(1):48, 2009.

[135] Vivian Hunt, Dennis Layton, and Sara Prince. Diversity matters. *McKinsey & Company*, 1:15–29, 2015.

[136] Lalit Jain, Kevin G Jamieson, and Rob Nowak. Finite sample prediction and recovery bounds for ordinal embedding. In *Advances In Neural Information Processing Systems*, pages 2703–2711, 2016.

[137] David G Jansson and Steven M Smith. Design fixation. *Design studies*, 12(1):3–11, 1991.

[138] Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[139] Tyler A Johnson, Avery Cheeley, Benjamin W Caldwell, and Matthew G Green. Comparison and extension of novelty metrics for problem-solving tasks. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T06A012–V007T06A012. American Society of Mechanical Engineers, 2016.

[140] Peter J Jordan, Neal M Ashkanasy, Charmine EJ Härtel, and Gregory S Hooper. Workgroup emotional intelligence: Scale development and relationship to team process effectiveness and goal focus. *Human Resource Management Review*, 12(2):195–214, 2002.

[141] Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.

[142] Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. In *Advances in Neural Information Processing Systems*, pages 6853–6863, 2017.

[143] Maryam Karimzadehgan and ChengXiang Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 1697–1700, 2009.

[144] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for online bipartite matching. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 352–358, 1990.

[145] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

[146] M.G. Kendall. *Rank correlation methods*. Theory and applications of rank order-statistics. Hafner Pub. Co., 1962.

[147] Trina C Kershaw and Stellan Ohlsson. Multiple causes of difficulty in insight: the case of the nine-dot problem. *Journal of experimental psychology: learning, memory, and cognition*, 30(1):3, 2004.

[148] Marcus Matthias Keupp and Oliver Gassmann. Determinants and archetype users of open innovation. *R&D Management*, 39(4):331–341, 2009.

[149] Joy Kim, Justin Cheng, and Michael S Bernstein. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 745–755. ACM, 2014.

[150] CF Kirschman, GM Fadel, and C Jara-Almonte. Classifying functions for mechanical design. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF MECHANICAL DESIGN*, 120:475–482, 1998.

[151] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.

[152] Mark Klein and Gregorio Convertino. A roadmap for open innovation systems. *Journal of Social Media for Organizations*, 2(1):1, 2015.

[153] Mark Klein and Ana Cristina Bicharra Garcia. The bag of stars: High-speed idea filtering for open innovation. 2014.

[154] Mark Klein and Ana Cristina Bicharra Garcia. High-speed idea filtering with the bag of lemons. *Decision Support Systems*, 78:39–50, 2015.

[155] Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet similarity comparisons. *stat*, 1050:28, 2016.

[156] Paul Kline. *The new psychometrics: science, psychology and measurement*. Routledge, 2014.

[157] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

[158] E Krissinel and K Henrick. Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, 60(12):2256–2268, 2004.

[159] Barry Matthew Kudrowitz and David Wallace. Assessing the quality of ideas from prolific, early-stage product ideation. *Journal of Engineering Design*, 24(2):120–139, 2013.

[160] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1193–1200, 2011.

[161] Alex Kulesza and Ben Taskar. Learning determinantal point processes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

[162] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.

[163] Ryoji Kurata, Masahiro Goto, Atsushi Iwasaki, and Makoto Yokoo. Controlled school choice with soft bounds and overlapping types. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[164] Vincent Lenhardt. *Coaching for meaning: The culture and practice of coaching and team building*. Insep Editions, 2004.

[165] Kevin E. Levay, Jeremy Freese, and James N. Druckman. The demographic and political composition of mechanical turk samples. *SAGE Open*, 6(1):2158244016636433, 2016.

[166] Daniel Levi. *Group dynamics for teams*. Sage Publications, 2015.

[167] Linjie Li, Vicente Malave, Amanda Song, and Angela J Yu. Extracting human face similarity judgments: Pairs or triplets? *Journal of Vision*, 16(12):719–719, 2016.

[168] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[169] Ulrich Lichtenthaler and Eckhard Lichtenthaler. A capability-based framework for open innovation: Complementing absorptive capacity. *Journal of Management Studies*, 46(8):1315–1338, 2009.

[170] Brian F Licuanan, Lesley R Dailey, and Michael D Mumford. Idea evaluation: Error in evaluating highly original ideas. *The Journal of Creative Behavior*, 41(1):1–27, 2007.

[171] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[172] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL-HLT)*, 2011.

[173] Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.

[174] Julie S Linsey, EF Clauss, Tolga Kurtoglu, JT Murphy, KL Wood, and AB Markman. An experimental study of group idea generation techniques: understanding the roles of idea representation and viewing methods. *Journal of Mechanical Design*, 133(3):031008, 2011.

[175] Julie Stahmer Linsey. *Design-by-analogy and representation in innovative engineering concept generation*. PhD thesis, University of Texas, Austin, 2007.

[176] Christian List. Lessons from the theory ofjudgment aggregation. *Collective wisdom: principles and mechanisms*, page 203, 2012.

[177] Xiang Liu, Torsten Suel, and Nasir Memon. A robust model for paper reviewer assignment. In *ACM Conf. on Recommender Systems (RecSys)*, 2014.

[178] Y-C Liu, A Chakrabarti, and T Bligh. Towards an 'ideal'approach for concept generation. *Design Studies*, 24(4):341–355, 2003.

[179] B Lopez-Mesa and R Vidal. Novelty metrics in engineering design experiments. In *DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia*, 2006.

[180] Miles Lubin, Emre Yamangil, Russell Bent, and Juan Pablo Vielma. Polyhedral approximation in mixed-integer convex optimization. *Mathematical Programming*, pages 1–30, 2016.

[181] Ioanna Lykourentzou, Faez Ahmed, Costas Papastathis, Irwyn Sadien, and Konstantinos Papangelis. When crowds give you lemons: Filtering innovative ideas using a diverse-bag-of-lemons strategy. In *21st ACM Conference on Computer-Supported Cooperative Work & Social Computing*, Jersey City, USA, November 2018. ACM.

[182] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[183] Mary Lou Maher and Douglas H Fisher. Using ai to evaluate creative designs. In *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity Volume 1*, 2012.

[184] A Majchrzak and A Malhotra. Viewpoint: Towards an information systems perspective and research agenda on crowdsourcing for innovation. *Journal of Strategic Information Systems*, 22:257–268, 2013.

[185] Christopher D Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*, volume 999. MIT Press, 1999.

[186] Elizabeth Mannix and Margaret A Neale. What differences make a difference? the promise and reality of diverse teams in organizations. *Psychological Science in the Public Interest*, 6(2):31–55, 2005.

[187] Daniel McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, pages S13–S29, 1980.

[188] Baharan Mirzasoleiman, Stefanie S Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. In *AAAI Conference on Artificial Intelligence 2018*. Association for the Advancement of Artificial Intelligence (AAAI), 2018.

[189] Ethan Mollick and Ramana Nanda. Wisdom or madness? comparing crowds with expert evaluation in funding the arts. *Management Science*, 62(6):1533–1553, 2015.

[190] William L. Moore. A cross-validity comparison of rating-based and choice-based conjoint analysis models. *International Journal of Research in Marketing*, 21(3):299 – 312, 2004.

[191] Pablo G Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 2015.

[192] Michael D Mumford and Sigrid B Gustafson. Creativity syndrome: Integration, application, and innovation. *Psychological bulletin*, 103(1):27, 1988.

[193] Brent A Nelson, Jamal O Wilson, David Rosen, and Jeannette Yen. Refined metrics for measuring ideation effectiveness. *Design Studies*, 30(6):737–743, 2009.

[194] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[195] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

[196] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

[197] Vo Dinh Minh Nhat, Duc Vo, Subhash Challa, and SungYoung Lee. Nonmetric mds for sensor localization. In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 396–400. IEEE, 2008.

[198] Besmira Nushi, Adish Singla, Anja Gruenheid, Erfan Zamanian, Andreas Krause, and Donald Kossmann. Crowd access path optimization: Diversity matters. In *Third AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2015.

[199] Sarah K Oman, Irem Y Tumer, Kris Wood, and Carolyn Seepersad. A comparison of creativity and innovation metrics and sample validation through in-class design projects. *Research in Engineering Design*, 24(1):65–92, 2013.

[200] Christian R Østergaard, Bram Timmermans, and Kari Kristinsson. Does a different view create something new? the effect of employee diversity on innovation. *Research Policy*, 40(3):500–509, 2011.

[201] Gerhard Pahl and Wolfgang Beitz. *Engineering design: a systematic approach*. Springer Science & Business Media, 2013.

[202] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[203] Michelle A Pang and Carolyn C Seepersad. Crowdsourcing the evaluation of design concepts with empathic priming. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering*

Conference, pages V007T06A004–V007T06A004. American Society of Mechanical Engineers, 2016.

[204] Ray Paramesh. Independence of irrelevant alternatives. *Econometrica (pre-1986)*, 41(5):987, 1973.

[205] GP Patil and Charles Taillie. Diversity as a concept and its measurement. *Journal of the American statistical Association*, 77(379):548–561, 1982.

[206] Linus Pauling and Barclay Kamb. *Linus Pauling: selected scientific papers*, volume 2. World Scientific, 2001.

[207] Paul B Paulus, Karen I van der Zee, and Jared Kenworthy. Cultural diversity and team creativity. In *The Palgrave Handbook of Creativity and Culture Research*, pages 57–76. Springer, 2016.

[208] John W. Payne, James R. Bettman, Eloise Coupey, and Eric J. Johnson. A constructive process view of decision making: Multiple strategies in judgment and choice. *Acta Psychologica*, 80(1):107 – 141, 1992.

[209] Jef Peeters, Paul-Armand Verhaegen, Dennis Vandevenne, and JR Duflou. Refined metrics for measuring novelty in ideation. *IDMME Virtual Concept Research in Interaction Design, Oct*, pages 20–22, 2010.

[210] Lisa Hope Pelled, Kathleen M Eisenhardt, and Katherine R Xin. Exploring the black box: An analysis of work group diversity, conflict and performance. *Administrative Science Quarterly*, 44(1):1–28, 1999.

[211] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[212] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems*, pages 2352–2360, 2016.

[213] Shameem A. Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 15–22, New York, NY, USA, 2016. ACM.

[214] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 15–22. ACM, 2016.

[215] Lena Qian and John S Gero. Function–behavior–structure paths and their role in analogy-based design. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 10(04):289–312, 1996.

[216] Lijing Qin and Xiaoyan Zhu. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[217] Stephen A Rhoades. The herfindahl-hirschman index. *Fed. Res. Bull.*, 79:188, 1993.

[218] Trevor Richardson, Brett Nekolny, Joseph Holub, and Eliot H Winer. Visualizing design spaces using two-dimensional contextual self-organizing maps. *AIAA Journal*, 52(4):725–738, 2014.

[219] Christoph Riedl, Ivo Blohm, Jan Marco Leimeister, and Helmut Krcmar. The effect of rating scales on decision quality and user attitudes in online innovation communities. *International Journal of Electronic Commerce*, 17(3):7–36, 2013.

[220] Alan G Robinson and Dean M Schroeder. *Ideas are free: How the idea revolution is liberating people and transforming organizations*. Berrett-Koehler Publishers, 2004.

[221] Alan G Robinson and Dean M Schroeder. *Ideas are free: How the idea revolution is liberating people and transforming organizations*. Berrett-Koehler Publishers, 2004.

[222] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 extended abstracts on Human factors in computing systems*, pages 2863–2872. ACM, 2010.

[223] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[224] Matthew J. Salganik and Karen E. C. Levy. Wiki surveys: Open and quantifiable social data collection. *PLOS ONE*, 10(5):1–17, 05 2015.

[225] Juho Salminen. *The role of collective intelligence in crowdsourcing innovation*. PhD thesis, Lappeenranta University of Technology, 2015.

[226] Swami Sankaranarayanan, Azadeh Alavi, and Rama Chellappa. Triplet similarity embedding for face verification. *arXiv preprint arXiv:1602.03418*, 2016.

[227] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890. ACM, 2010.

[228] Prabir Sarkar and Amaresh Chakrabarti. Assessing design creativity. *Design Studies*, 32(4):348–383, 2011.

[229] Kai Sassenberg, Kai J Jonas, James Y Shah, and Paige C Brazy. Why some groups just feel better: The regulatory fit of group power. *Journal of Personality and Social Psychology*, 92(2):249, 2007.

[230] Charles E Schaefer and Anne Anastasi. A biographical inventory for identifying creativity in adolescent boys. *Journal of Applied Psychology*, 52(1p1):42, 1968.

[231] Candice Schumann, Samsara N. Counts, Jeffrey Foster, and John P. Dickerson. The diverse cohort selection problem. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2019.

[232] Suzanne Scotchmer and Jerry Green. Novelty and disclosure in patent law. *The RAND Journal of Economics*, pages 131–146, 1990.

[233] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. A framework for recommending relevant and diverse items. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[234] Jami J Shah, Santosh V Kulkarni, and Noe Vargas-Hernandez. Evaluation of idea generation methods for conceptual design: effectiveness metrics and design of experiments. *Journal of Mechanical Design*, 122(4):377–384, 2000.

[235] Jami J Shah, Steve M Smith, and Noe Vargas-Hernandez. Metrics for measuring ideation effectiveness. *Design studies*, 24(2):111–134, 2003.

[236] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[237] David Shatz. *Peer review: A critical inquiry*. Rowman & Littlefield, 2004.

[238] Pao Siangliulue, Kenneth C Arnold, Krzysztof Z Gajos, and Steven P Dow. Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 937–945. ACM, 2015.

[239] Edward H Simpson. Measurement of diversity. *Nature*, 163(4148):688, 1949.

[240] Wouter Sluis-Thiescheffer, Tilde Bekker, Berry Eggen, Arnold Vermeeren, and Huib De Ridder. Measuring and comparing novelty for design solutions generated by young children through different design methods. *Design Studies*, 43:48–73, 2016.

[241] Steven M Smith, Thomas B Ward, and Jay S Schumacher. Constraining effects of examples in a creative generation task. *Memory & cognition*, 21(6):837–845, 1993.

[242] Elizabeth Starkey, Christine A Toh, and Scarlett R Miller. Abandoning creativity: The evolution of creative ideas in engineering design course projects. *Design Studies*, 47:47–72, 2016.

[243] Elizabeth M Starkey, Samuel T Hunter, and Scarlett R Miller. Are creativity and self-efficacy at odds? an exploration in variations of product dissection in engineering education. *Journal of Mechanical Design*, 141(1):012001, 2019.

[244] Robert J Sternberg. *Handbook of creativity*. Cambridge University Press, 1999.

[245] Neil Stewart, Gordon DA Brown, and Nick Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

[246] Andy Stirling. A general framework for analysing diversity in science, technology and society. *Journal of the Royal Society Interface*, 4(15):707–719, 2007.

[247] Robert B Stone and Kristin L Wood. Development of a functional basis for design. *Journal of Mechanical Design*, 122(4):359–370, 2000.

[248] Kazunari Sugiyama and Min-Yen Kan. Scholarly paper recommendation via user's recent research interests. In *Conference on Digital Libraries*, pages 29–38, 2010.

[249] James Surowiecki. The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business. *Economies, Societies and Nations*, 296, 2004.

[250] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, 1988.

[251] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 673–680, USA, 2011. Omnipress.

[252] Omer Tamuz, Ce Liu, Ohad Shamir, Adam Kalai, and Serge J. Belongie. Adaptively learning the crowd kernel. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 673–680, New York, NY, USA, 2011. ACM.

[253] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

[254] Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *Conference on Web Intelligence and Intelligent Agent Technology (WIC)*. IEEE, 2010.

[255] CW Taylor and RL Ellison. Alpha biographical inventory. *Salt Lake City, UT: Institute for Behavioral Research in Creativity*, 1966.

[256] Christine A Toh and Scarlett R Miller. Choosing creativity: the role of individual risk and ambiguity aversion on creative concept selection in engineering design. *Research in Engineering Design*, 27(3):195–219, 2016.

[257] Warren S Torgerson. Theory and methods of scaling. 1958.

[258] E Paul Torrance. Predictive validity of the torrance tests of creative thinking. *The Journal of creative behavior*, 6(4):236–262, 1972.

[259] Olivier Toubia and Laurent Florès. Adaptive idea screening using consumers. *Marketing Science*, 26(3):342–360, 2007.

[260] Michele Twomey, Lee A Wallis, and Jonathan E Myers. Limitations in validating emergency department triage scales. *Emergency Medicine Journal*, 24(7):477–479, 2007.

[261] Antti Ukkonen, Behrouz Derakhshan, and Hannes Heikinheimo. Crowdsourced nonparametric density estimation using relative distances. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[262] Eliot van Buskirk. Google struggles to give away 10 million. http://www.wired.com/2010/06/google-struggles-to-give-away-10-million/all/1, 2010. Archived on: 2016-02-04.

[263] L. van der Maaten and K. Weinberger. Stochastic triplet embedding. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, Sept 2012.

[264] Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.

[265] Jeroen J. G. van Merrienboer, Paul A. Kirschner, and Liesbeth Kester. Taking the load off a learnerś mind: Instructional design for complex learning. *Educational Psychologist*, 38(1):5–13, 2003.

[266] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 109–116, New York, NY, USA, 2011. ACM.

[267] P-A Verhaegen, Dennis Vandevenne, and JR Duflou. Originality and novelty: a different universe. In *DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia*, 2012.

[268] Paul-Armand Verhaegen, Dennis Vandevenne, Jef Peeters, and Joost R Duflou. Refinements to the variety metric for idea evaluation. *Design Studies*, 34(2):243–263, 2013.

[269] S. V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *J. Mach. Learn. Res.*, 11:1201–1242, August 2010.

[270] Eric Von Hippel. Democratizing innovation: The evolving phenomenon of user innovation. *Journal für Betriebswirtschaft*, 55(1):63–78, 2005.

[271] Thomas Wagenknecht, Jan Crommelinck, Timm Teubner, and Christof Weinhardt. When life gives you lemons: How rating scales affect user activity and frustration in collaborative evaluation processes. In *13th International Conference on Wirtschaftsinformatik*, feb 2017.

[272] Xiaojie Wang, Zhicheng Dou, Tetsuya Sakai, and Ji-Rong Wen. Evaluating search result diversity using intent hierarchies. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 415–424, New York, NY, USA, 2016. ACM.

[273] Michael Wilber, Iljung S. Kwak, David Kriegman, and Serge Belongie. Learning concept embeddings with combined human-machine expertise. In *International Conference on Computer Vision (ICCV)*, 2015.

[274] Michael Wilber, Iljung S Kwak, David Kriegman, and Serge Belongie. Learning concept embeddings with combined human-machine expertise. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 981–989, 2015.

[275] Katherine Y Williams and Charles A O'Reilly III. Demography and. *Research in Organizational Behavior*, 20:77–140, 1998.

[276] Jamal O Wilson, David Rosen, Brent A Nelson, and Jeannette Yen. The effects of biological examples in idea generation. *Design Studies*, 31(2):169–186, 2010.

[277] Anbang Xu and Brian Bailey. A reference-based scoring model for increasing the findability of promising ideas in innovation pipelines. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1183–1186, New York, NY, USA, 2012. ACM.

[278] Teng Ye and Lionel P Robert Jr. Does collectivism inhibit individual creativity?: The effects of collectivism and perceived diversity on individual creativity and satisfaction in virtual ideation teams. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 2344–2358. ACM, 2017.

[279] Q. Yu, E. L. Xu, and S. Cui. Submodular maximization with multi-knapsack constraints and its applications in scientific literature recommendations. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1295–1299, Dec 2016.

[280] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Sketch-a-net that beats humans. *arXiv preprint arXiv:1501.07873*, 2015.

[281] Qilian Yu, Easton Li Xu, and Shuguang Cui. Streaming algorithms for news and scientific literature recommendation: Submodular maximization with a d-knapsack constraint. *arXiv preprint arXiv:1603.05614*, 2016.

[282] Mehmet Ersin Yumer, Paul Asente, Radomir Mech, and Levent Burak Kara. Procedural modeling using autoencoder networks. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 109–118. ACM, 2015.

[283] Cheng Xiang Zhai, William W Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 10–17. ACM, 2003.

[284] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. Improving web search results using affinity graph. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2005.

[285] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511. ACM, 2005.

[286] Mi Zhang and Neil Hurley. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 508–515, Washington, DC, USA, 2009. IEEE Computer Society.

[287] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 13–22. ACM, 2012.

[288] Pengfei Zhao and Dik Lun Lee. How Much Novelty is Relevant? It Depends on Your Curiosity. In *39th International ACM SIGIR Conference on Research and Development, Pisa, Italy*, page 100, 2016.

[289] Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104. Citeseer, 2007.

[290] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.