THE UNIVERSITY OF CHICAGO

SEMI-AUTOMATED TOOLING FOR FILE MANAGEMENT IN PERSONAL CLOUD STORAGE

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
WILL BRACKENBURY

CHICAGO, ILLINOIS
APRIL 27TH, 2022

Dedicated to Shadow, who made this much more fun, and much harder.

"O for a Muse of fire, that would ascend / The brightest heaven of invention..."

-Henry V, Act I, Prologue

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Many thanks to all those who made this dissertation possible. To my parents, for your support, your encouragement, and for your patience with all my complaining. To Elyse, for more love and support than I could have asked for. To Andrew, for everything: I would not have graduated without you. To Taha, for getting me started and keeping me going. To Jean, for your friendship and counsel. To Brian, for your help and your patience. To Tyler, for your readiness to lend a hand. To John, for your time and expertise. To Taylor, for making research something I always looked forward to. To Ashwin, for believing in me, and sparking the fire that led to this. To Kyle, for giving time you did not have to spare in order to help.

To my research collaborators, Andrew (again), Galen, Yuxin, Chenhao, Michael L., Michael F., Rui, Mainack, Jillian, Jason, Weijia, Guan, Jamar, Kwam, Kevin, Abhi, and Zechao. You made this happen. To my thesis committee, for their attention and commitment. And lastly, to my advisers, Drs. Ur and Elmore, for pushing me.

ABSTRACT

In this dissertation, we investigate file collections in personal cloud storage and design semi-automated tools to support file management in that setting. We do so in three main investigations. In the first, we examine research participants' perceptions of file pairs in their Google Drive accounts. In the second, we conduct two online user studies asking participants to organize their Google Drive accounts in order to investigate real-time file management and evaluate the tool we developed to assist file management, KondoCloud. In the last, we proposed and evaluated a new format for summarizing groups of file management recommendations. Throughout our investigations, we find that developing more complex support for file management is generally feasible. We conclude by discussing the implications for future designs of file management tools.

CHAPTER 1

INTRODUCTION

The setting of personal cloud storage (e.g., Google Drive, Dropbox) is a fertile ground for investigations into personal information management (PIM). File collections that were previously challenging to study are now more readily available. While some software is available to study file collections on local storage at scale (Dinneen et al. [2016]), recruiting participants can be challenging. As such, many investigations can only be carried out in special circumstances, such as with corporate access to file repositories (Agrawal et al. [2007]). In cloud storage, though, participants are able to more easily share access to their file collections, opening a range of possibilities for research.

This is especially fruitful because cloud storage further offers challenges both familiar and novel. For example, difficulties in organizing files (Malone [1983]) and retrieving files (Boardman and Sasse [2004], Whittaker et al. [2010], Bergman et al. [2012]) persist in the cloud, but challenges such as managing shared files (Khan et al. [2018], Voida et al. [2013], Massey et al. [2014a]) and deleting useless or privacy-sensitive files (Khan et al. [2021], Clark et al. [2015]) are new. Techniques drawn from tools built for local storage (Liu et al. [2018], Sinha and Basu [2012], Bao and Dietterich [2011]) can be tested in new environments, while tools built for the cloud to help retrieve files (Tata et al. [2017], Xu et al. [2020]) or save files to preferable locations (Bergman et al. [2019]) may offer new modes of interaction to investigate. We compare against prior work further in Chapter 2.

This dissertation focuses on investigating file collections in personal cloud storage, and designing semi-automated tools to support file management in that arena. While prior work has investigated user perceptions of and behavior in cloud storage (Massey et al. [2014a], Voida et al. [2013]), it has minimally explored the file collections themselves. Given that the file collections in cloud storage may differ from those in local storage, understanding them prior to developing tools for the cloud is necessary. Here, we focus in particular on participant

perceptions of *similarity* between files in cloud storage, and how this may connect with user desires to manage files in similar ways (Chapter 3). We examined this in the first of four total user studies we conducted, the Investigation Study, where we recruited 50 participants to report on their perceptions of file pairs in their Google Drive accounts. We found that participants had many files they perceived as similar located in very different parts of their file hierarchy. Further, they often wished to manage these files in similar ways.

With this understanding, we developed and evaluated a tool for personal cloud storage, *KondoCloud*, that, like celebrity organizer Marie Kondo, helps to reduce messiness (Chapter 4). It does so by offering recommendations of file movement and deletion actions that users may wish to take in their file collection, based on actions they have taken on similar files. Given that such file management actions are not so easily reversible, we avoid the fully-automated method of taking these actions on a user's behalf without asking first. This tool improves on the state of the art as the first tool to fully support file management recommendations beyond retrieval, allowing users to iteratively improve the underlying disorganization in their file collection. We evaluated this tool in an online user study, finding that nearly half of participants using the full KondoCloud interface accepted a non-trivial portion ($> 10\%$) of offered recommendations, and a few accepted nearly all of them ($> 90\%$).

To improve on drawbacks identified in KondoCloud, we then proposed methods to succinctly summarize groups of file management recommendations (Chapter 5). Though prior work has sought to improve explanations of single recommendations in this setting (Xu et al. [2020], Gedikli et al. [2014]), it has not generalized this to multiple recommendations. Because presenting groups of related recommendations individually can mask context and burden users, developing summaries to collapse groups of recommendations into a single recommendation is a potential benefit for future systems. We evaluated the structure of these summaries in a within-subjects online user study, and found that these summaries displayed a number of beneficial properties when compared to baselines.

To conclude this dissertation, we discuss in Chapter 6 what these lessons mean for future work, and how the state of the art might be further improved.

CHAPTER 2

RELATED WORK

2.1 File Organization Behavior

Researchers first studied file organization in offices Lansdale [1988], Malone [1983], Kwasnik [1989], later analyzing digital analogues Barreau [1995b], Barreau and Nardi [1995], Boardman and Sasse [2004]. These studies developed several frameworks that describe how humans categorize documents and files. Malone [1983] and Kwasnik [1989] asked eight and ten participants, respectively, to describe the organization of their office spaces, and Barreau [1995b] performed a similar study on seven managers' digital file collections. Kwasnik and Barreau used these studies to develop frameworks describing how humans classify documents and files, and Bergman et al., in two separate studies (Bergman et al. [2003, 2008a]) investigated whether such a framework (the "User-Subjective Approach") could drive the development of tools for managing file collections.

Beyond frameworks for categorization, researchers have previously investigated high-level views of the actions people take when organizing information. Researchers have described the "foraging"-like behavior that characterizes information acquisition (Pirolli [2007], Belkin [1980], Kuhlthau [1991]), and the subsequent "keeping" (Jones et al. [2002]) and "curation" behaviors (Whittaker [2011], Oh [2012]) involved in retaining files for later retrieval (Cockburn and McKenzie [2001], Aula et al. [2005], Alrashed et al. [2018]). Researchers have also studied why later retrieval is difficult (Whittaker et al. [2010], Oh and Belkin [2014], Boardman et al. [2003], Mackenzie et al. [2019]), and how the choice of operating system (Bergman et al. [2012], Dinneen and Frissen [2020]) and organization of files (Bergman et al. [2010]) influence this. Several studies have also looked more directly at file interaction in personal cloud storage. Jahanbakhsh et al. [2020] investigated users' recognition and interest in files based on how recently a file was last accessed, as well as the richness of prior interactions,

and Xu et al. [2020] also explored the potential for recommender systems in cloud storage based on the type of interaction users previously had with the files. Our work contrasts with these two by offering recommendations at the time that an organizational action is taken, instead of at the start of a new organizational session.

2.2 Organization-Adjacent Tools

Researchers have prototyped a number of partial to fully automated tools to help users handle disorganized files and emails. Some tools assisted with disorganization by attempting to assist with non-navigation based file retrieval. These tools fall into two domains: those that passively assist users, and those that required active intervention. In the former domain, several tools have provided automatic shortcuts to files or emails of interest (Liu et al. [2018], Bao et al. [2006], Bao and Dietterich [2011]) or highlighted content likely to be accessed (Fitchett et al. [2014], Sen et al. [2021], Tran et al. [2016], Lee and Bederson [2003], Rhodes and Starner). These tools were either based on simple heuristics, or activity monitoring (Voida and Mynatt [2009]). The tools developed on this dissertation improve on these by offering recommendations beyond file retrieval, and by using more file and metadata based features. Other tools that passively assist users are interfaces that eschew typical folder-based organization. Such examples are Lifestreams' chronological display of information artifacts (Freeman and Gelernter [1996]), Confluence's time-based contextual retrieval (Gyllstrom [2009]), and "concept maps" that organize information using a hierarchy of topics (Yang et al. [2012]). In contrast to this passive assistance, systems like Haystack (Quan et al. [2003]), Stuff I've Seen (Dumais et al. [2003]), and various Semantic Desktop tools (Schröder et al. [2019], Chirita et al. [2006], Sauermann et al. [2006]) enhance an interface's search capability to improve re-finding. This requires an active effort on the part of the user: later work has identified the potential drawback of such methods as requiring additional cognitive overhead (Teevan et al. [2004], Bergman et al. [2008b, 2013c]).

Other tools aided organization behaviors directly. In the domain of file management, Bergman et al. [2009] developed GrayArea, which provides a "deletion-lite" option. In the same context of cloud storage we investigate, Bergman et al. developed a tool that nudged participants to save files in their cloud storage to a suggested folder (Bergman et al. [2019]), similar to a tool in local storage investigated by Sinha and Basu [2012]. Researchers have also built tools to aid in the organization of other types of collections. For example, Segal and Kephart [1999]'s MailCat suggests appropriate folders for an email. Other tools group emails by topic (Cselle et al. [2007]) or by additional features (Tang et al. [2008]). In the context of collections of bookmarks, information about bookmarks' social context can aid organization and discovery (Abrams et al. [1998], Millen et al. [2007, 2006]). None of these tools, however, offer either ongoing organizational support beyond the first "save" action, nor do they help with extant disorganization. More similar to the work in this dissertation is FileWeaver, a tool that automatically tracks and propagates dependencies between files (e.g., capturing the relationship between a script and the files that it generated) in an enhanced file management interface (Gori et al. [2020]). While this tool performs some semi-automated organization, it focuses solely on files with dependencies, whereas the tools in this dissertation make more general recommendations.

2.3 Alternatives to Folder-based Organizing

While occasional claims are made to the contrary (Mason and Seltzer [2019], Whitham and Cruickshank [2017]), folder-based organization remains the most common method of interacting with file collections. However, disadvantages, such as the lack of multiple categorization (Albadri et al. [2016], Sajedi et al. [2012]), have led to some alternative interface types that this dissertation draws lessons from. For example, while information re-finding could rely purely on search features, prior studies have found that users prefer standard file-management interfaces for navigation and re-finding (Bergman et al. [2008b, 2013a]).

Multiple researchers (Teevan et al. [2004], Bergman et al. [2008b, 2013c]) conducted studies with semi-structured interviews, longitudinal measurement, and in-lab studies that identified a few reasons why navigation is preferred over search. They found that search has a higher cognitive burden, and forming a search query requires a user to recall some context for the file without any aid. Teevan et al. [2004] found that users navigate through file hierarchies using additional context gained at each step of navigation. We do not investigate search-related behavior, but this knowledge of users' navigation through file hierarchies informs our investigation of file hierarchy structure. Related to search, Civan et al. [2008] and Bergman et al. [2013b] found that relying on file tags for information retrieval posed similar difficulties to search because tagging leaves files "placeless". The tools in this dissertation correspondingly offer users spatial context about files whenever relevant.

2.4 Scale & Structure of File Hierarchies

There are many prior investigations of file collections' scale and structure (Dinneen and Julien [2019], Oh [2017]). Researchers recruited anywhere from a handful (3) to many thousands ($\sim 60,000$) participants from universities, single companies, and the general population (Henderson and Srinivasan [2009], Hardof-Jaffe et al. [2009], Agrawal et al. [2007], Zhang and Hu [2014], Dinneen et al. [2019]). The studied file collections, though, were almost all from local storage. Hardof-Jaffe et al. [2009], however, studied online storage of 2,081 undergraduate students at Tel-Aviv university. Our descriptions of file collection scale differ from this because the samples in each of our studies are drawn from crowdworkers on commercial cloud storage, which is potentially a population more representative of users we wish to support with our tools. Many works also taxonomize the structure of file hierarchies. We compare between prior works and our own in Table 2.1. Like with models of file categorization, the overlap is not perfect. Some taxonomies, such as the one from Oh [2017], focus on participant behavior, while others, such as Hardof-Jaffe et al. [2009]'s focus on the structure

Ours	Malone [1983]	Boardman and Sasse [2004]	Hardof-Jaffe et al. [2009]	Henderson and Srinivasan [2009]	Vitale et al. [2018]	Oh [2017]
Flat	Pilers		Piling / One-Folder Filing	Pilers	Hoarding	Fuzzy
		Extensive Filers / Occasional Filers	Big Folders Filing	Filers		Flexible
Deep	Filers	Total Filers	Small Folders Filing	Structurers	Minimalist	Rigid

Table 2.1: A comparison of file hierarchy scale / structure taxonomies

of the file hierarchy itself. Our own taxonomy, introduced in Section 3.4, concentrates on the latter, and aligns well with prior work.

2.5 Interfaces for Information Management

This dissertation also draws lessons from work in adaptive interfaces, given this sub-field's relevance to our design of KondoCloud (Chapter 4). Greenberg and Witten first identified the potential for interfaces that rearrange in response to user activity (Greenberg and Witten [1985]). Sears and Shneiderman [1994] expanded on this approach by limiting the reordering to only occur above a "split" in the menu. KondoCloud consists of a non-adaptive component that resembles standard file browsers, augmented by recommendations that change in response to user activity. Gajos et al. [2006] studied this broad kind of "split interface" approach in a 26-participant lab study. Their participants were more satisfied with the split interface than alternatives, which the authors attributed to the interface's *spatial stability*, the property that menu items have a base location where they can always be located. KondoCloud shares this trait. Other user studies identified predictability, accuracy, and feature awareness as important traits in user satisfaction with adaptive interfaces (Gajos et al. [2008], Findlater and McGrenere [2010]). KondoCloud abides by these principles.

2.6 Set Summarization

Similar to how we summarize sets of recommendations in Chapter 5, researchers have summarized sets of items in numerous ways. Some techniques summarize with a representative subset of the items, such as centroid approaches (Likas et al. [2003]), top-k (Chaudhuri and

Gravano [1999]), regret minimization (Kessler Faulkner et al. [2015]), KL-divergence (Yan et al. [2005]), maximum entropy (Wang and Parthasarathy [2006]), or Bayesian Information Criterion (Mampaey et al. [2011]). We avoid such techniques, given their low **verifiability**, one of the key criteria upon which we develop our own set summaries, defined in Chapter 5. Other techniques extract feature information to generate a plaintext summary, as in text summarization (Yao et al. [2017]), or image captioning (Hossain et al. [2019]). These summaries, however, are also unlikely to be verifiable, and are generated via a training set of existing summaries, which are not available for our domain. Alternatively, researchers have used application-specific visualizations to represent the item space (Joglekar et al. [2017], Chen and Hung [2009]). These visualizations, however, require global consistency across different summaries, while we do not. This allows for more succinct summaries that are more efficient to synthesize. Similar work that has visually represented local summaries has not been generalized to the setting of multiple recommendations (Ribeiro et al. [2016]). We borrow parts of these prior works by incorporating a hover interaction into our visual summaries (*Decision Tree* and *Rules-Tree*, Chapter 5) that shows what files are covered by a predicate of the summary.

More closely related to our techniques are summaries using tables of characteristics (Wen et al. [2018], El Gebaly et al. [2014]). Our rules-based summaries extend these by also generating predicates over set-typed data. Similar to summary tables, associative rules for frequent itemsets (Agrawal et al. [1993], Borgelt [2012]), and their related techniques for classification (Dong et al. [1999], Li et al. [2001]) seek to generate and describe relationships over related items. These techniques, like the visual explanations described above, require global consistency.

It is less common for set summarization to have been applied to the domain of recommender systems. The closest analogues are in conversational recommender systems, where some researchers summarize how the set of unexplored items differs from the set of explored

items (Chen and Pu [2007]). Researchers have augmented this to describe categories of unexplored items based on extracted review sentiment (Chen and Wang [2017]). Other work, while it does not investigate summaries, has focused on related sets of recommendations, which it dubs "slate" recommendations (Mehrotra et al. [2019], Swaminathan et al. [2017]). The summaries we develop in Chapter 5 can therefore be interpreted as seeking methods to summarize these slates.

2.7 AI Explanations

The recommendations summaries we develop in Chapter 5 generalize explanations in AI systems (Guidotti et al. [2018b], Tjoa and Guan [2020], Danilevsky et al. [2020], Adadi and Berrada [2018], Narayanan et al. [2018], Kim et al. [2016], Lipton [2018]). Explanations have been shown to improve users' understanding (Sinha and Swearingen [2002]) and trust (Dzindolet et al. [2003]) in a system and help teach users when a system can be relied upon to make accurate judgments (Lapuschkin et al. [2019], Ross et al. [2017]). Many explanation types are based on more "interpretable" models, such as sparse linear classifiers (Ribeiro et al. [2016]), rule sets (Wang et al. [2017], Wang and Rudin [2015]), trees (Lou et al. [2012]), or programs (Singh et al. [2016]). Our proposed summaries bear a strong resemblance to rule set explanations. We adapt these to the setting of file recommendation and make two improvements: we do not require pre-mining predicates, and we present our explanations in plaintext (Chang et al. [2016]). The predicates in our summaries also resemble short programs (e.g., a Python function, as in Singh et al. [2016]). Given that the target users of our summaries are non-technical, though, we avoid programming syntax. We compare directly against decision-tree-based explanations (Lou et al. [2012]) in our online study, as these are a proxy for many "intrepretable" models. As noted by Lipton [2018], the interpretability of such models may be overstated–we discuss this in Section (5.2). While other works have augmented interpretable models in ways that compare closely to our own

work (Guidotti et al. [2018a]), we differ from these in our generation of set-based predicates (also described in Section 5.2).

Researchers have also studied explanations specifically in recommender systems (Tintarev and Masthoff [2015], Zhang et al. [2020]), KondoCloud being one example of such systems. Beyond verifiability, known also as "scrutability" (Tintarev and Masthoff [2012], Czarkowski and Kay [2002]) or "simulatibility" (Lipton [2018]), prior work has proposed other evaluation metrics for explanations in recommender systems. Some are based on users' perceptions of explanations, such as transparency (Sinha and Swearingen [2002]), or their improvement in users' overall satisfaction with a system (Tanaka-Ishii and Frank [2000]). Others are task-based, such as an explanation's ability to justify a recommendation (Vig et al. [2009]), to help users hone in on their preferences (Bilgic and Mooney [2005]), to enhance users' understanding of available items (Felfernig and Gula [2006]), to persuade them (Cramer et al. [2008], Herlocker [2000]), or to increase their decision-making speed (McCarthy et al. [2004]). Of these, the ability to help users hone in on preferences, and enhance understanding of available items, are most relevant to the evaluation of our recommendation group summaries in Chapter 5.

CHAPTER 3

FILE SIMILARITY IN CLOUD STORAGE

3.1 Overview

Users spend a significant amount of time viewing, curating, and organizing collections of digital files (Whitham and Cruickshank [2017], Gao [2011], Whittaker [2011]). Building on the success of recommender systems in other contexts (Covington et al. [2016], Bennett et al. [2007], Smith and Linden [2017], Zheng et al. [2018]), researchers have developed a number of recommender systems to help users identify files they wish to retrieve (Tata et al. [2017], Chen et al. [2020], Jahanbakhsh et al. [2020], Fitchett et al. [2014], Liu et al. [2018]). Prior user-centered research has found that abstract notions of file similarity underpin how users view file organization and retrieval (Malone [1983], Barreau [1995b], Bergman et al. [2003], Boardman and Sasse [2004]). It is no surprise, then, that these recommender systems implicitly seem to rely on file similarity to make recommendations for file retrieval. For example, in Google Drive, if a user edits a document, the Quick Access tool (Tata et al. [2017], Chen et al. [2020]) may suggest other files that were last modified at similar times.

However, these recent systems take a relatively narrow view of what it means for files to be similar. For instance, while most systems concretize similarity in terms of access patterns, we hypothesize that similarity of metadata and content features (e.g., filenames, objects recognized in images) might provide important signals to recommender systems. Furthermore, previous work focused almost exclusively on file retrieval, leaving open the question of whether a system that observes a user deleting or moving a file should also recommend that they delete or move (to the same place) similar files. We use the term **co-management** to describe this broader pattern of *managing similar files in similar ways*.

In this chapter, we answer a series of complementary questions about conceptualizing co-management and file similarity more broadly than in prior work by conducting a two-

part, online user study of 50 Google Drive and Dropbox users and their cloud accounts (the Investigation Study). The first part surveyed participants about how they used and organized their cloud accounts. After receiving participant consent, we also used the Google Drive and Dropbox APIs to analyze participants' accounts, collect metadata, and compute the similarity of pairs of files in the account in terms of eleven metadata and content features. Once this automatic processing had concluded, the participant returned for the second part of the study, answering survey questions about how they perceived the similarity between 18 pairs of files from their account, as well as whether they wanted to co-manage those files (i.e., find, move, or delete them together).

As our first research question, we wondered to what degree seemingly curated file repositories in consumer cloud storage stand to benefit from recommender systems. In Section 3.5, we examine the structure of participants' cloud accounts, focusing on where similar files are located. Echoing prior work (Malone [1983], Hardof-Jaffe et al. [2009], Henderson and Srinivasan [2009], Vitale et al. [2018], Oh [2017]), we found that some participants piled most of their files into a small number of folders (termed a **piler** hierarchy), while others organized their files into many folders with long chains of subfolders (termed a **filer** hierarchy). Our first key contribution came from analyzing the relative locations of pairs of files perceived by participants to be similar in these hierarchies. Intuition might have suggested that similar files would be located in the same directory, or perhaps in an adjacent directory. However, we found this not to be the case. Even in superficially organized filer hierarchies, pairs of files participants perceived as similar were located far away in the directory structure. We observed a similar result when looking at files' automatically extractable metadata and content features. As such, even the types of users whom prior work characterized as organized filers stand to benefit from automated recommendations about files that are inconvenient to find or that have been forgotten.

Our second research question was whether participants actually wanted to co-manage

files they perceived as similar. In Section 3.6, we present correlations in our survey results, showing that participants did indeed desire to co-manage the majority of files they perceived as similar, whereas they wanted to co-manage only a small fraction of files they did not perceive as similar. Whereas existing tools (Bao et al. [2006], Fitchett et al. [2014], Tata et al. [2017], Fitchett and Cockburn [2012], Liu et al. [2018]) already leverage this result for finding and retrieving files, we show similar results for co-moving and co-deleting similar files, highlighting the need for broader co-management recommendations than are currently provided.

To lay the foundation for transitioning these insights to tools, our third research question investigated what metadata and content features are predictive both of whether humans perceive files as similar and whether they want to co-manage them. Existing tools focus on temporal information, such as files' last modification date or last access time, as a proxy for similarity (Bao et al. [2006], Fitchett et al. [2014], Tata et al. [2017], Fitchett and Cockburn [2012], Liu et al. [2018]). While, as detailed in Section 3.7, our regression models did find temporal information to be predictive of both perceived similarity and desired co-management, we also identified metadata features (e.g., the similarity of filenames) and content features (e.g., the similarity of words used in a document or of objects recognized in images) as predictive. We conclude the chapter in Section 3.8 by discussing how these insights are incorporated into the tools and techniques described in Chapters 4 and 5.

3.2 Framework and Definitions

Here, we define our notions of perceived similarity, data similarity, and co-management in the context of prior work. Toward building richer tools for information management, we empirically evaluate and quantify relationships among these concepts in Sections 3.6–3.7.

Ours	Kwasnik [1992], Barreau [1995b]	Bergman et al. [2003]	Boardman and Sasse [2004]
Topic	Document Attributes	Subjective Classification Principle	Topic
Creation Context	Situation Attributes / Document Attributes / Time	Subjective Context Principle	
Derivation		Subjective Context Principle	
Purpose	Disposition / Situation Attributes	Subjective Context Principle	
	Order / Scheme		
	Document Attributes		Document Class
	Value	Subjective Importance Principle	

Table 3.1: Comparison of our framework for perceived similarity (left) with notions of similarity discussed in prior work.

3.2.1 Perceived Similarity

We define *perceived similarity* as a user's subjective perception about how files may be similar or dissimilar. For example, users may *perceive* two documents to be similar if they were written by the same author or describe the same project. Prior work describes how many people use this idea to describe the organization of their files (Kwasnik [1989], Barreau [1995b], Bergman et al. [2003], Boardman and Sasse [2004]) and organize them for later retrieval (Jones et al. [2005], Whittaker [2011]).

To evaluate whether users wish to manage similar files similarly, we focus on four dimensions of perceived file similarity synthesized from prior work (Kwasnik [1989], Barreau [1995b], Bergman et al. [2003], Boardman and Sasse [2004]). Table 3.1 summarizes differences between our framework and prior work. Our framework includes:

- **Topic:** Two files are similar if they are about the same subject. Kwasnik [1992]'s and Barreau [1995b]'s frameworks described this concept as part of "Document Attributes," which included other items like "Author" and "Physical Form" (e.g., a spreadsheet printout). Topic also falls under Bergman et al. [2003]'s "Subjective Classification Principle" (information with the same subject should be categorized together). **Example**: a photo of a dog and a document about dog grooming.

- **Purpose:** Two files are similar if they will likely be used for similar tasks or purposes. Purpose is a subset of "Situation Attributes" in Kwasnik [1992]'s and Barreau [1995b]'s frameworks, but also includes aspects of "Disposition," a user's intentions

about whether to keep or discard the file. Bergman et al. [2003]'s "Subjective Context Principle" also encompasses Purpose, as Purpose is part of the context when a file is saved. **Example**: a receipt and a W-2 form both saved for tax calculations.

- **Derivation:** Two files are similar if they are different versions of the same item, or if one "created" the other. Derivation is included under Bergman et al. [2003]'s "Subjective Context Principle" given that a version of an item contains the same implicit context. **Example**: a paper outline and the final version of that paper.

- **Creation context:** Two files are similar if they were created at the same time, by the same person, or in the same place. Kwasnik [1992]'s and Barreau [1995b]'s frameworks separate this across several categories as sub-attributes of "Source" ("Situation Attributes"), "Author" ("Document Attributes"), and "Time." **Example**: a poem authored at a writer's retreat and another person's poem written at the same retreat.

For three reasons, our framework does not include the attributes "Order" / "Scheme" (e.g., grouping, arrangement), "Document Attributes" (e.g., color, size), or "Value" (e.g., important, needs improvement) defined in other frameworks (Kwasnik [1991], Bergman et al. [2003], Boardman and Sasse [2004]). First, in an empirical study, these aspects were some of the least common ways that interviewees described their file collections (Kwasnik [1991]). Second, "Document Attributes" and "Document Class" can more naturally be considered data similarity (defined later in this section), rather than perceived similarity. Third, "Order" and "Scheme" describe the organizational structure of a file collection, not perceived similarity. We use this synthesis of past work to guide our investigation of the relationship between similarity and co-management. Expanding or critically reevaluating past frameworks is not our focus. Prior work has investigated the reliability of these framework components, finding them to correspond to how users describe similarity (Kwasnik [1991]).

Feature	Files	Description
Time		
Last Modified	All	Logarithm of difference, in seconds, between the two files' last modified dates
Metadata		
Filename	All	Jaccard similarity of the list of bigrams (two-letter chunks) in the filenames
File Size	All	Logarithm of difference, in bytes, between the file size
Tree Distance	All	The number of steps to reach one file from the other when traversing the file hierarchy (represented as a tree)
Shared Users	All	Jaccard similarity of the lists of unique user IDs with whom the files have been shared
Contents		
File Contents	All	Jaccard similarity of chunks of the raw file contents using MinHash
Text Contents	Text	Cosine similarity between documents' Word2Vec (Mikolov et al. [2013]) vector embeddings
Text Topic	Text	Cosine similarity of documents' Term Frequency Inverse Document Frequency (TF-IDF) vectors (Wu et al. [2008])
Table Schema	Spreadsheets	Jaccard similarity of the column names of spreadsheets, such as .xlsx, .csv, and .tsv files
Image Contents	Images	Jaccard similarity between unique objects recognized in images by object-detection algorithms (Google [2019])
Image Color	Images	Absolute difference between the average RGB values of each image

Table 3.2: The data similarity features we examine in the user studies described in this dissertation, the files to which they apply, and how we computed them. We cluster these features in three groups: time (the focus of the most closely related work from Fitchett et al. [2014], Liu et al. [2018] and Tata et al. [2017]), file metadata, and file contents.

3.2.2 Data Similarity

We define **data similarity** as comparisons of features that can be algorithmically extracted from files without human intervention. These features include **time** (e.g., last modified time), **metadata** (e.g., filename and size), and **content** features (e.g., text topics and objects identified in images). Table 3.2 lists all data similarity features we considered within these three categories. We hypothesized that data similarity could, at scale, help identify files perceived as similar.

Prior work has postulated that data similarity can be used to identify similar items (Quan et al. [2003], Oppermann et al. [2020], Canuto et al. [2019]), yet did not fully test these claims. Prior implementations (Bao et al. [2006], Fitchett et al. [2014], Tata et al. [2017], Fitchett and Cockburn [2012], Liu et al. [2018]) have focused almost exclusively on time features, such as file-access patterns or recently accessed files. That said, tools like Haystack (Quan et al. [2003]) do use text features for retrieval, yet they do so in the context of a user-defined query, rather than by comparing files. The seminal Remembrance Agent (Rhodes and Starner), which recommends other files that might be relevant, is most similar to how we envision the use of data similarity. However, the Remembrance Agent only uses

text features. In short, we explore more diverse and comprehensive features than prior work.

3.2.3 Co-management

We refer to the pattern of managing similar files similarly as **co-management**. Supporting a user's ability to co-manage files has the potential to passively improve a user's file organization over time, similar to tools that identify the best folders for a user to save a new file or email (Bergman et al. [2019], Sinha and Basu [2012], Segal and Kephart [1999]). We consider the following actions:

- **Find:** If a user accesses a file, they may also want to access another similar file.
- **Move:** If a user moves a file to another folder, they may also want to move another similar file to the same folder.
- **Delete:** If a user deletes a file, they may also want to delete another similar file.

We focused on Find, Move, and Delete actions because they are commonly studied and used in practice. The Find action relates to prior work that used recent file or folder accesses to provide shortcuts to similar files or folders (Bao et al. [2006], Fitchett et al. [2014], Tata et al. [2017], Fitchett and Cockburn [2012], Liu et al. [2018]). We did not investigate actions that are less common or less foundational for information management, such as renaming, creating symlinks, or copying files (Bergman et al. [2013b], Oh [2012], Dinneen and Julien [2019]). Future work could expand our co-management framework to evaluate those strategies.

3.3 Investigation Study Methodology

To answer our research questions, we conducted a two-part online user study (the Investigation Study). In Part 1, we asked participants about file management abstractly and

performed an automated scan of their cloud account. In Part 2, we elicited participants' perceptions of the similarity between, and desire to co-manage, 18 pairs of files from their account. Section 7.1 contains our full survey instrument, and Section 7.2 contains our full regression tables.

3.3.1 Recruitment and Part 1 Survey

We recruited participants on Prolific [2019], a recommended alternative (Peer et al. [2017]) to the Amazon Mechanical Turk crowdsourcing marketplace. We required they be age 18+, live in the USA, and have completed 100+ tasks with 95% approval. We also required participants to have a Google Drive or Dropbox account that was at least three months old and had at least 100 files, including one shared file. In the short Part 1 survey that followed, we asked general questions about participants' demographics and organization of their cloud account. This portion took 15 minutes on average. Compensation was $2.50.

3.3.2 File Processing

Once the participant authorized access to their cloud account, we used the Google Drive or Dropbox API to analyze their account, collect file metadata, and compute data similarity features. We extracted text from documents, as well as column headers from data tables. Using the Google Vision API (Google [2019]), we also computed a color histogram, listed recognized objects, and extracted available text from images. To reduce computational costs, we only collected data similarity features pairwise on a stratified sample of 1000 files whose distribution of file types matched the underlying account's. For confidentiality, we hashed all human-readable information with a participant-specific salt that we discarded after processing.

Once processing was complete, we selected 18 pairs of files to show participants in Part 2. For each of the following criteria, we randomly chose pairs from all files which satisfied

the criterion.

- 2 pairs had similar filenames (based on their bigrams)
- 2 pairs' filenames had a small Levenshtein edit distance
- 2 pairs had a similar set of shared users
- 2 pairs had a similar text topic (based on TF-IDF, from Wu et al. [2008])
- 2 pairs had a similar table schema
- 2 pairs had similar image contents (in Google Vision, from Google [2019])
- 1 pair was in the same directory (tree distance 0)
- 1 pair was located at tree distance 1
- 4 pairs were selected randomly

We added additional random pairs whenever an insufficient number of files matched any criterion above. Thresholds were set via pilot testing. Due to a coding error, the tree distance of some file pairs was calculated incorrectly during sampling, inadvertently excluding a small number of file pairs that otherwise might have been selected based on being in the same directory or at tree distance 1. This error was corrected prior to our data analysis and would only have impacted sampling for a few files matching a corner case.

3.3.3 Part 2

Once we finished processing a participant's files, we invited them to Part 2, a survey centered on these 18 pairs of files from their own account. For each file pair, in randomized order, we first asked the participant to describe both files in free text. We then asked them to describe in free text how they believed the files were similar or dissimilar. Next, we asked

them to rate their agreement with a series of statements on five-point Likert scales ("strongly agree" to "strongly disagree," plus a "don't know" option). This series included statements about our four classes of perceived similarity (e.g., "I consider these two files to be similar in *Topic*"). It also included statements about our three types of co-management (e.g., "If I were searching for information, and I found one of these files to be relevant, I would also want to *see* the other file"). Part 2 took approximately one hour to complete. Compensation was $10.00.

3.3.4 Analysis Approach

We analyzed three types of data: (i) general metadata for all files in each participant's account; (ii) data similarity features computed pairwise for a representative sample of 1,000 files in each participant's account; and (iii) detailed survey responses from participants about 900 file pairs (50 participants $\times$ 18 pairs each). We report illustrative quotes from participants, but do not formally analyze them qualitatively. We used the 900 labeled file pairs to build mixed-effects ordinal logistic regression models with the four types of perceived similarity and three types of co-management as our dependent variables. Because the data was not independent, we included a random effect for each participant. The data similarity features were our independent variables. When a given data similarity feature was not applicable (e.g., the Image Contents feature does not apply when comparing a spreadsheet and an image), or in the rare cases when our extractor encountered an error (e.g., reading a malformed file), we filled missing values as 0 or 1 for similarity and distance features, respectively.

3.3.5 Limitations

We report on a convenience sample of crowdworkers that is not representative of any broader population. Despite efforts to communicate how our data collection respected the privacy of participants' accounts, privacy-conscious crowdworkers were unlikely to participate, further

biasing our sample. Because we asked the same questions for each file pair, participants may have been prone to fatigue and inattention (Lavrakas [2008]). We mitigated this concern by iteratively shortening both multiple-choice and free-response sections through extensive pilot testing, as well as restricting the study to 18 file pairs. We chose to investigate personal file collections in cloud accounts because of the uniform and comprehensive APIs that Google Drive and Dropbox provide. Past work has noted that cloud accounts represent only part of a user's fragmented file collection (Capra and Perez-Quinones [2006]), so our results may not generalize to other types of file collections. Notably, the types of files present, the organizational structure, and the usage context may all differ in local storage. Lastly, asking participants sequentially about perceived similarity and desired co-management may have biased them to identify similarity or co-management when they would not have done so otherwise. Future work should build on the lessons learned to investigate perceived similarity and co-management in a more naturalistic setting.

3.4 Participants and Their Accounts

Here, we describe our participants and their cloud accounts.

3.4.1 Participant Demographics

In total, 50 participants completed the Investigation Study protocol. Among participants, 54.0% were female, 40.0% were male, and 6.0% were non-binary. The most common age range was 25–34 years old (48.0%) and the second most common was 18–24 years old (30.0%). Among participants, 26.0% had held a job or taken a course in computer science. For the study, 92.0% of participants used Google Drive, while 8.0% used Dropbox. Most participants (98.0%) reported using their service's web app to access their account. 60.0% reported using a mobile app, and 30.0% reported having automatic sync enabled. Participants reported being daily (34.0%), weekly (50.0%), or monthly users (14.0%) of their account; one participant

	Min	Q_1	Median	Q_3	Max
Age of oldest file (days)	148	2,405	3,001	3,570	4,546
Total size (GB)	< 1	2	5	11	151
Total # files	123	298	541	1,445	17,081
(# images)	0	33	168	732	15,123
(# documents)	4	44	140	298	2,345
(# spreadsheets)	0	5	16	34	207
(# presentations)	0	0	3	10	152
(# web files)	0	0	0	2	2,453
(# media files)	0	6	41	129	5,532
(# other files)	0	7	25	87	10,060
Total # folders	3	9	27	95	3,185
Unique file extensions	5	12	15	24	75

Table 3.3: Characteristics of participants' cloud accounts.

chose not to respond. On average, participants estimated that their account contained 74.6% personal data and 25.4% professional data.

3.4.2 Participants' Cloud Accounts

Table 3.3 reports general characteristics of participant accounts. Our 50 participants collectively stored 119,388 files in their accounts. The median account was 8 years old and contained 5 gigabytes of data. Across accounts, we observed 341 unique file extensions. The most common file type was images (72,125 files), mostly .jpg (46,019) and .png (22,422) files. Second was a catch-all "other" category (14,729). The most common "other" file extension was flat (3,664), which is for database files, with .json (580) and .zip (402) as next most common. Documents (13,405) and media files (12,334) followed. Following trends observed in prior work on file collections (Dinneen et al. [2019]), many characteristics were lognormally distributed, causing a large gap between the 75th percentile (Q_3) and the maximum value. We therefore report an adjusted mean (e^{μ}, in Dinneen et al. [2019]) where appropriate. Due to sampling differences, we leave a comparison against the scale and structure of file collections in local storage to future work.

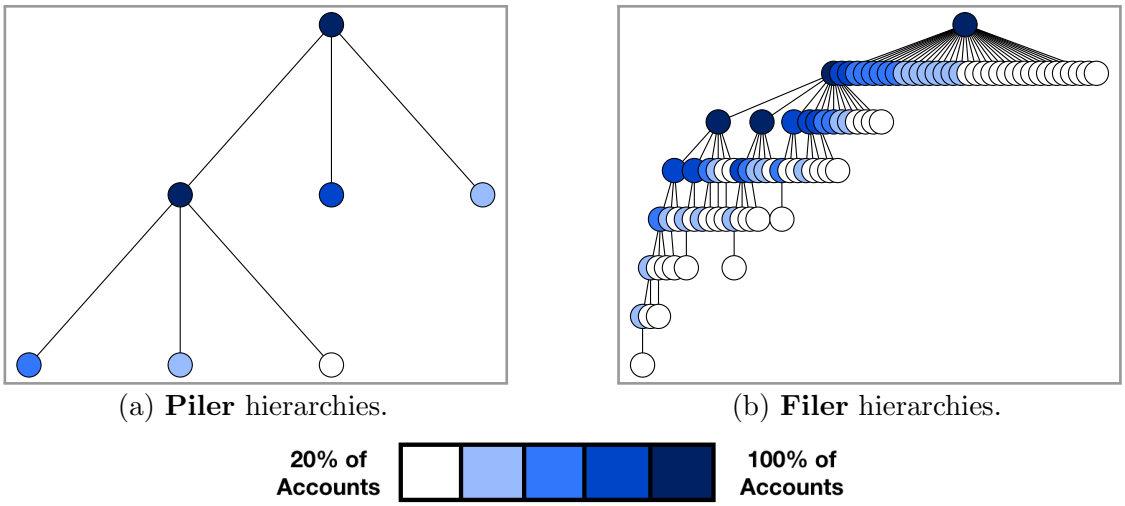
Participants were split on whether they considered their account well-organized: 36.0%

	Filer	Piler
e^{μ} # files	1,146	311
e^{μ} # folders	75	6
Mean files per folder	25	69
Mean depth	3.07	1.01
Mean breadth	9.13	2.37
e^{μ} # unique file extensions	26.00	11.00
e^{μ} # unique folders per extension	8.44	1.32
e^{μ} # unique extensions per folder	1.86	4.13

Table 3.4: Comparison of account characteristics by hierarchy type. e^{μ} is the adjusted mean of the distribution. *Depth* is the number of clicks needed to reach a file from the root. *Breadth* is the number of subfolders in a folder.

reported that their account is well-organized, 40.0% disagreed, and 24.0% were neutral. Many well-organized participants justified their self-perception with their usage of folders (*"I name the folder of the topic what the photos or files fall under."*). They also reported strategies like organizing files by date. In contrast, some disorganized participants chose not to use folders (*"With text search and picture view I find it irrelevant"*), or reported difficulties doing so (*"I have folders I use to split up files, but I threw everything up there... and I have to go back and reorganize it"*).

We examined the scale and structure of participants' accounts, as well as participants' free-text responses concerning organization, and found that participants' accounts naturally split into two groups matching those found in prior work (Malone [1983], Boardman and Sasse [2004], Hardof-Jaffe et al. [2009], Henderson and Srinivasan [2009], Vitale et al. [2018], Oh [2017]). Participants had 50 folders on average, with a standard deviation of 531.9, while the median participant had 27 folders. Combining k-means clustering on the number of folders per participant with free-text responses yielded a cluster threshold of 10. We thus term accounts containing 10 or fewer folders **pilers** and accounts with over 10 folders **filers**. Among participants, 28.0% were pilers, while 72.0% were filers. Figure 3.1 visualizes the typical folder hierarchy for both classes. Each node in the tree represents a folder, colored proportional to the percentage of accounts that contained such a folder. We pruned all nodes



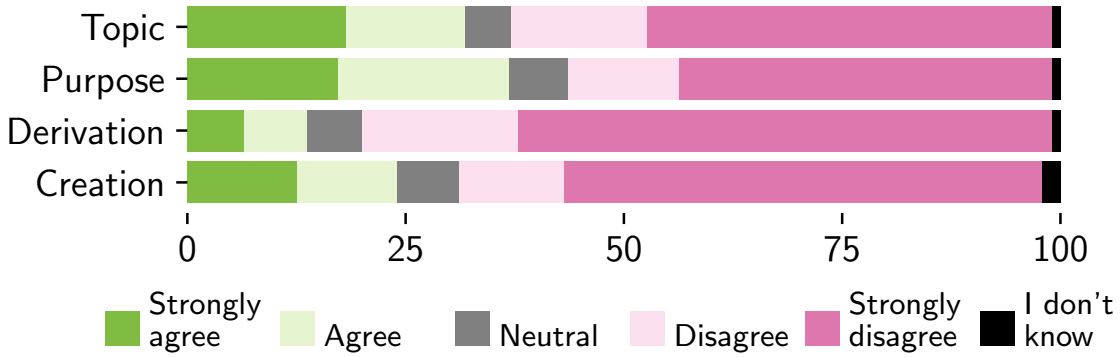
(a) **Piler** hierarchies. (b) **Filer** hierarchies.

Figure 3.1: The typical folder structure of piler and filer hierarchies. These trees merge participants' file structures, coloring nodes by the percentage of participants with that hierarchy type who had a node at that location. Nodes appearing for $< 20\%$ of participants with that hierarchy were pruned.

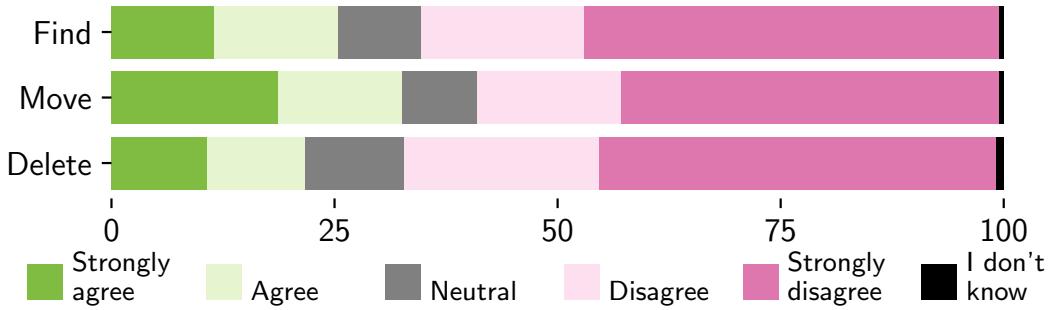
that appeared in under 20% of accounts. As shown in Figure 3.1, piler hierarchies typically contained the root directory and one or two sub-folders. In contrast, most filer hierarchies contained many sub-folders and a few deeper branches. Table 3.4 further quantifies differences between piler and filer hierarchies. We investigate how these differences in hierarchy relate to similarity and co-management in Sections 3.5–3.6.

3.5 Account Organization

In this section, we present participant's overall responses about the perceived similarity and desired co-management of file pairs. We also investigate how pairs of files that participants perceived as similar, pairs of files that appeared similar in terms of data similarity features, and pairs of files that participants wanted to co-manage were distributed in the file hierarchy. If files were organized tightly by similarity, files that are similar would be located in the same folder, and files that are not similar would be located in different folders. We observed, however, that files that were similar in both participant perception and data characteristics, as well as files that participants wanted to co-manage, were distributed throughout the file



(a) Perceived similarity.



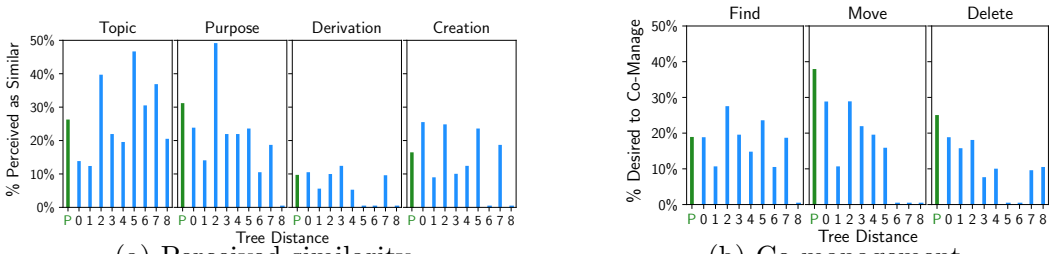
(b) Co-management

Figure 3.2: Participants' agreement that file pairs exhibited the four types of perceived similarity (top) or that the files should be co-managed in each of the three ways (bottom).

hierarchy. This result highlights the need for recommender systems to help users co-manage files in cloud accounts.

3.5.1 Analysis of Responses Overall

Figure 3.2a displays the distribution of participants' responses for the perceived similarity of the 900 file pairs they labeled. Each response was on a five-point Likert scale. Participants perceived file pairs as similar (responded "strongly agree" or "agree") in at least one of the four dimensions 46.9% of the time. Among our similarity dimensions, participants most often perceived pairs as similar in purpose (36.9% of pairs). Note that our stratified sampling approach was purposely biased to identify more similar file pairs. Among only the 417 file pairs selected randomly, participants perceived 39.0% as similar in at least one dimension,



(a) Perceived similarity (b) Co-management

Figure 3.3: The percentage of files participants perceived as similar (left) or desired to co-manage (right) either in piler (P) hierarchies or broken out by tree distance (*numbers*) in filer hierarchies. Both "strongly agree" and "agree" responses indicate similarity or co-management here. So that the percentages are meaningful, we only consider file pairs selected either randomly or based on tree distance, not those selected based on having similar features.

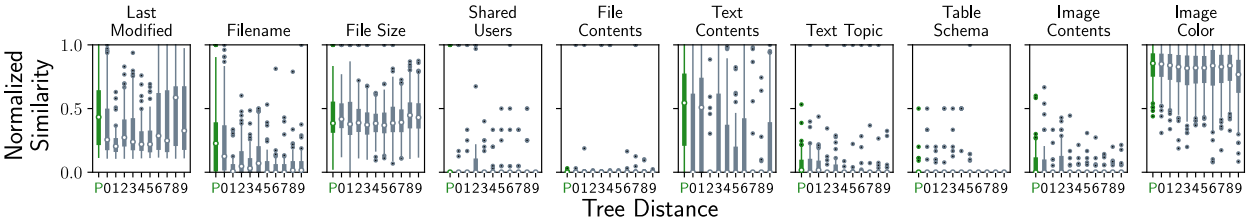


Figure 3.4: Box plots depicting how each class of **data similarity** is distributed for all file pairs in participants' accounts. The box plot labeled P shows the distribution for all pairs in piler accounts. The remaining box plots represent the distribution in filer accounts at the tree distance specified by the label (e.g., "0" represents the distribution for file pairs in the same directory).

and 29.7% as similar in purpose. This proportion is likely closer to the underlying distribution. We also note that perceived similarity differs significantly by dimension, ranging between 13.8% for derivation to 36.9% for purpose.

Figure 3.2b displays the distribution of participants' ratings about their desire to co-manage the 900 file pairs. The trends in perceived similarity hold here as well. Participants infrequently wanted to co-manage files, and the rates at which they did varied by the type of co-management. For randomly selected file pairs, participants desired to find, move, or delete files together for 19.7%, 26.6%, and 15.8% of file pairs, respectively, less than in our stratified sample.

3.5.2 Perceived Similarity in the File Hierarchy

Surprisingly, files that participants perceived as similar were often found in very different parts of the file hierarchy. Many of our analyses are based on **tree distance**, or the minimum number of transitions (to parent or child folders) to get from one folder to the other. Files in the same folder have tree distance 0, while files in adjacent folders have tree distance 1.

Figure 3.3a shows the distribution of file pairs perceived as similar with respect to tree distance in both piler and filer hierarchies. In filer hierarchies, 46.6% of file pairs that participants perceived as similar in at least one dimension had tree distance > 2, and 19.5% had tree distance ≥ 5. Participants frequently described files located far apart in the file hierarchy as very similar. For instance, for a file pair with tree distance 13, a participant wrote, *"These files are very similar. They are both songs that I like, by artists I like. They are a similar genre."* For all four types of perceived similarity, at least 92.5% of pairs at tree distance 2 were in "sibling" folders (i.e., the files' parent folders share the same parent). This organization pattern was described in prior work by Dinneen et al. [2019] and Teevan et al. [2004] as a technique users employ to gradually filter into more fine-grained categories.

Because our stratified sampling targeted file pairs more likely to be similar than a random file pair, Figure 3.3 likely overestimates file similarity. We therefore examined the subset of file pairs that were sampled either randomly or only based on tree distance, finding similar trends. Of the file pairs sampled in this way that were perceived as similar in at least one dimension, 39.7% had Tree Distance > 2, while 19.8% had Tree Distance ≥ 5. In sum, we found that similar files are often not located in the same folder, and they are sometimes located quite far in the file hierarchy.

3.5.3 Co-management in the File Hierarchy

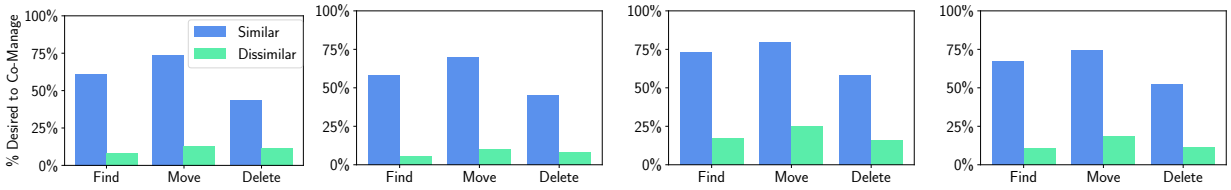
Figure 3.3b shows similar trends in participants' desire to co-manage files at different tree distances. Of file pairs that participants wanted to co-manage (find, move, or delete to-

gether), 40.0%, 37.7%, and 36.6%, respectively, had tree distance > 2. Of these files, 17.1%, 10.6%, and 14.9%, respectively, had tree distance ≥ 5. A participant described the similarity between files they wanted to move together (despite a tree distance of 7) as, *"They are both trainings but we need to keep them by month for our grant."* We also examined only the file pairs that were selected randomly, finding similar trends.

3.5.4 Data Similarity in the File Hierarchy

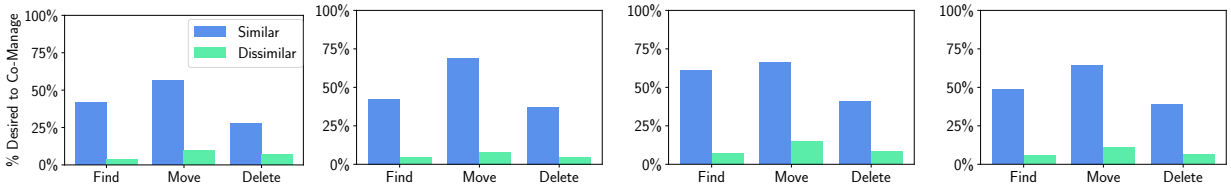
Finally, we explored the relationship between data similarity features and tree distance. We analyzed 11,653,450 pairs of data similarity features for all file types, with an additional 4,519,675 pairs for image similarity features, 4,262,444 for text similarity features, and 39,333 for table similarity. To our knowledge, this is the first large-scale analysis of data similarity in cloud storage. Figure 3.4 shows the relationship between data similarity and tree distance. We make two key observations. First, many more file pairs are dissimilar than are similar. Second, tree distance does not appear to correlate strongly with data similarity features. Intuitively, if users categorize files within a file hierarchy with similar files close to each other in the hierarchy, then one would expect to see median similarity decrease with tree distance. This does not occur here.

Taken together, these analyses emphasize that files that participants perceive as similar, files that participants wish to co-manage, and files that look similar in terms of algorithmically extractable features are all often located far apart in the file hierarchy. Potential explanations for the phenomenon itself include the following: the existence of distinct, but overlapping file hierarchies (Jones et al. [2005], Boardman et al. [2003]); the desire of users to categorize a file in multiple ways, but choosing one by necessity of the interface (Bergman et al. [2013a]); and the existence of partially categorized files (Oh [2012]). We leave further investigation of root explanations to future work. Regardless, the dispersed locations of similar files will inhibit future retrieval without improved tools.



(a) Topic (b) Purpose (c) Derivation (d) Creation Context

Figure 3.5: How participants' desire to co-manage files correlated with their perceptions of files as similar in one of our four dimensions of perceived similarity. We binned "strongly agree" and "agree" responses as similar / to be co-managed.



(a) Topic (b) Purpose (c) Derivation (d) Creation Context

Figure 3.6: This figure is the same as Figure 3.5, but considers only "strongly agree" responses for similarity/co-management.

3.6 Similarity Implies Co-Management

We found that a file pair's perceived similarity strongly correlated with whether a participant wished to co-manage it. Specifically, participants wished to co-manage similar file pairs at a much higher rate than dissimilar pairs. Because participants expressed perceptions of similarity and desire to co-manage on 5-point Likert scales, we tried binarizing their preference in two ways: based on both strong and mild preferences ("strongly agree" and "agree" responses indicated similarity/co-management, Figure 3.5) or only on strong preferences (only "strongly agree" responses, Figure 3.6). Comparing the figures, the correlation between similarity and co-management held regardless of preference strength. For example, among file pairs perceived as similar in creation context (strong and mild), participants wanted to co-move 74.7% of them, whereas they only wanted to co-move 18.4% of dissimilar file pairs. For strong preferences only, participants desired to co-move 64.6% of similar pairs, versus 11.1% of dissimilar pairs. The relationship between similarity and co-management was statistically

	Perceived Similarity				Co-management		
	Topic	**Purpose**	**Derivation**	**Creation**	**Find**	**Move**	**Delete**
Data Similarity							
Last Modified	20.769***	11.422***	3.207**	17.075***	12.034***	12.618***	7.623***
Filename	3.978**	12.699***	18.094***	12.805***	10.885***	13.744***	4.286***
File Size	0.985	1.718	1.829	1.507	1.379	1.656	2.231*
Tree Distance	0.471	0.588	0.863	1.496	0.409	0.500	1.090
Shared Users	2.428**	2.874***	2.857**	3.124***	6.833***	7.218***	5.604***
File Contents	2.855**	3.473***	4.172***	3.703***	2.536**	2.027*	2.197**
Text Topic	3.072***	2.592**	2.315*	2.260*	1.707	2.588**	1.526
Table Schema	3.965	13.076*	1.845	2.393	2.905	1.951	3.993
Image Contents	36.777***	29.018***	8.938***	8.757***	13.085***	10.106***	2.767
Filer Hierarchy	0.884	1.255	1.078	2.036	1.385	0.704	0.789
Random Effects							
σ of random effect	1.095	0.680	1.519	1.140	1.409	1.251	1.416

Table 3.5: Our regression models showing odds ratios for data similarity features (*** $p < .001$; ** $p < .01$; * $p < .05$).

significant regardless of similarity or co-management type (Spearman's rank correlation test, all $p < 0.001$). In the remainder of this dissertation, we binarize based on both strong and mild preferences unless stated otherwise.

However, correlation between similarity and co-management was not perfect; participants also wished to co-manage some dissimilar file pairs. Among pairs that participants wished to co-find, 23.6%, 60.3%, 15.3%, and 36.2% were dissimilar in topic, derivation, purpose, and creation context, respectively. Some were similar in another dimension (*"Same student, but the content is much different"*), but many were explicitly dissimilar (*"They are dissimilar because File 1 is for dissertation and File 2 is for my job"*).

Overall, this evidence suggests that co-management tools based on perceived similarity and informed by data similarity might be able to identify files participants wish to co-manage and would not naturally discover.

3.7 Modeling Based on Data Similarity

While the previous section highlighted the connection between perceived similarity and co-management, this insight is difficult to act on because perceived similarity is a "human" value. Thus, we built regression models to correlate algorithmically extractable features (data similarity) with perceived similarity and desired co-management. We found several features to be highly predictive.

Table 3.5 gives the odds ratios for our logistic regressions. These coefficients can be interpreted as the multiplicative increase in the probability that the response variable will be one level higher (e.g., "agree" to "strongly agree") for an increase of 1 in the data similarity value. All of our data similarity values are normalized to a $[0, 1]$ scale, and all distance metrics are turned into similarity metrics by subtracting their distance from the maximum value of 1. Therefore, the odds ratio is the multiplicative increase if a value has full similarity in that dimension, versus none.

Some features, such as similarities in last modified times, are known to be predictive (Bao et al. [2006], Fitchett et al. [2014], Tata et al. [2017], Fitchett and Cockburn [2012], Liu et al. [2018]). Others, such as image contents and filename, have rarely been used. Shared users, file contents, and text topic features were also statistically significant, but with smaller effect sizes. That tree distance was not a significant predictor matches evidence from prior sections. We also found no significant effect for whether a hierarchy was a piler or filer, suggesting that the importance of data similarity features may hold across both types of hierarchies. The size of the random effects indicates that individual variations between participants accounted for approximately half a point change in the mean Likert-scale rating of file pairs. This result suggests that user-specific features (e.g., personality / mood, from Massey et al. [2014b] and Whittaker and Massey [2020]) may affect perceived similarity.

Many factors that were predictive of perceived similarity were also predictive of co-management. One exception was the image contents feature, which was not predictive of

co-deletion, though this may be an artifact of our sample size. Future tools should leverage these features' predictiveness in supporting co-management.

3.8 Summary

We investigated whether similarity can support co-management via an online study of 50 Google Drive and Dropbox users (the Investigation Study). We found that similar files were distributed across the file hierarchy, and that a user's perception of similarity between two files correlated with their desire to co-manage those files. We explored through regression analysis the ability of data similarity to predict perceived similarity and co-management. Last Modified, Image Contents, Filename, Shared Users, and Text Topic features were significant.

We extract four design principles from this work that inform the design of KondoCloud (Chapter 4):

- Recommendations must work beyond retrieval. The demonstrated links between similarity and co-movement / co-deletion suggest that tools supporting such behaviors can offer utility to end-users. KondoCloud supports these actions.

- Recommendations must work across the hierarchy. Participants wanted to co-manage files located both close and far in the file hierarchy. Previous tools, such as Fitchett et al. [2014]'s enhanced finder interface, only highlighted file or folder icons in the current folder. Our results show that this leaves significant functionality untouched; users might overlook files in other folders. KondoCloud, therefore, uses adaptive split-screen interface enhancements like in Liu et al. [2018] to offer recommendations on files in folders besides the one currently being viewed.

- Recommendations must work for both Piler and Filer hierarchies. As in the previous point, highlighting icons would likely be inappropriate in a piler hierarchy. There

are likely to be many files in a single folder, and highlighted icons would not be sufficiently visible or provide context. On the other hand, in filer hierarchies, files are likely to be further apart, and it would be important to provide context on where co-managed files live (e.g., showing a visualization of the file hierarchy). Tools implementing co-management must support both types of contextual feedback. We did not find a significant impact of hierarchy type on the rate at which recommended actions were taken in KondoCloud, suggesting that the results potentially hold independent of hierarchy type.

- Recommendations must use features beyond access patterns. Access patterns are a highly informative feature. In fact, many prior studies and tools restrict the scope of recommendations to recently accessed files (Bergman et al. [2012, 2010], Tata et al. [2017]). However, users have difficulty retrieving older or infrequently accessed files (Whittaker et al. [2010]), which are of interest to users (Jahanbakhsh et al. [2020]). Access patterns are unlikely to be an informative feature for these files. We use richer content-based features for the classifiers in KondoCloud in order to address this drawback of prior tools. The effect size of these features is lower than access pattern features (Table 4.2), and summaries using these features are less commonly generated (Figure 5.2). We nonetheless find these to be of greater importance than assigned previously.

CHAPTER 4

KONDOCLOUD

4.1 Overview

Numerous existing tools (Liu et al. [2018], Tata et al. [2017], Dumais et al. [2003], Fitchett et al. [2014]) help users retrieve files of interest from within disorganized personal file collections, including cloud repositories. However, these tools do not attempt to address the underlying disorganization. Researchers have developed prototype interfaces and tools that take alternate approaches beyond the standard file-and-folder paradigm (Voida and Mynatt [2009], Gyllstrom [2009], Marsden and Cairns [2004], Dourish et al. [2000], Dourish [2003]), but these tools have seen limited adoption, potentially due to users' strong preference for navigating to files through a folder hierarchy (Bergman et al. [2008b], Jones et al. [2005]). The few tools working over folder hierarchies that do try to help users organize their data, in contexts ranging from cloud repositories to emails (Bergman et al. [2019], Segal and Kephart [1999], Sinha and Basu [2012]), only attempt to aid in the organization of data that has not yet been added to the repository. They do not aim to help users organize data that has already accumulated there. Given this limited support from existing tools, it is unsurprising that users organize infrequently (Boardman and Sasse [2004]).

To help users organize their personal cloud repositories, we designed *KondoCloud*, a file-browser interface that, like its namesake (celebrity organizer Marie Kondo), reduces clutter. It does so by providing machine-learning-based recommendations of files the user might want to move, delete, or retrieve. These recommendations leverage the idea of co-management, as introduced in Chapter 3. This idea is concretized in Figure 4.1: if, for example, a user moves a given file to a folder, KondoCloud may suggest moving other, similar files to that same folder.

To inform KondoCloud's design, we first conducted an online user study, the *Observa-*

tion Study, in which we asked 69 crowdworkers to spend 30 minutes organizing their Google Drive repositories in a standard file-browsing interface while we logged their actions. To our knowledge, this is the first empirical examination of users' real-time organizational strategies in cloud repositories. We identified several high-level strategies, including moving files to newly created folders, extensively deleting files, and re-categorizing misplaced files into existing folders. To concretize the notion of file similarity that underpins KondoCloud's recommendations, participants labeled the similarity between pairs of files, also indicating whether they wanted to manage the files in similar ways. In addition, we collected pairwise data similarity features (from Table 3.2 in Chapter 3) for the file pairs. From this labeled data, we trained logistic regression classifiers to predict pairs of files that should be managed similarly. Each classifier achieved an F1 score of at least 0.72, which is appropriate for human-in-the-loop recommendations.

Using this classifier and our new knowledge of organizational strategies, we designed and built KondoCloud, our file-browsing user interface with embedded recommendations. Figure 4.7 in Section 4.4 presents the KondoCloud user interface. As previously mentioned, KondoCloud uses our classifier to make recommendations for files the user might want to move, delete, or retrieve based on having performed the same action on a similar file in the past.

We evaluated KondoCloud and the recommendations it makes in a between-subjects online user study, the *Evaluation Study*. We randomly assigned 59 participants to use the KondoCloud interface either with or without the recommendations generated by our classifier while organizing their own Google Drive repository. Nearly half of participants who saw recommendations accepted some of the recommendations, and a few accepted almost all of them. KondoCloud's recommendations helped participants delete related files that were spread across different directories. Further, many recommendations captured actions the user hoped to take. In a follow-up survey, participants strongly agreed (on a Likert

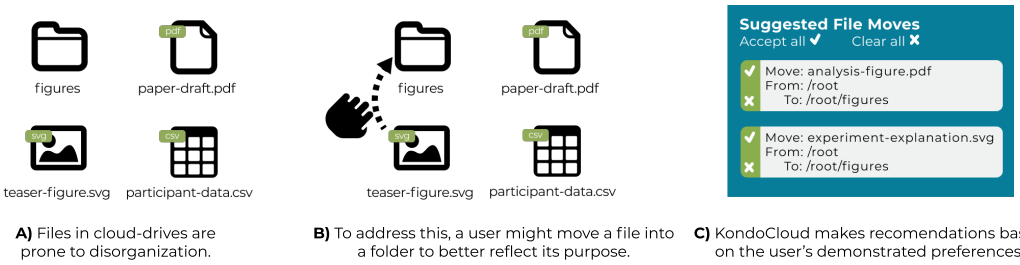


Figure 4.1: KondoCloud is a file-browsing interface that helps users organize cloud repositories (e.g., Google Drive) by providing ML-based recommendations for files they may want to move, delete, or retrieve based on past actions on similar files.

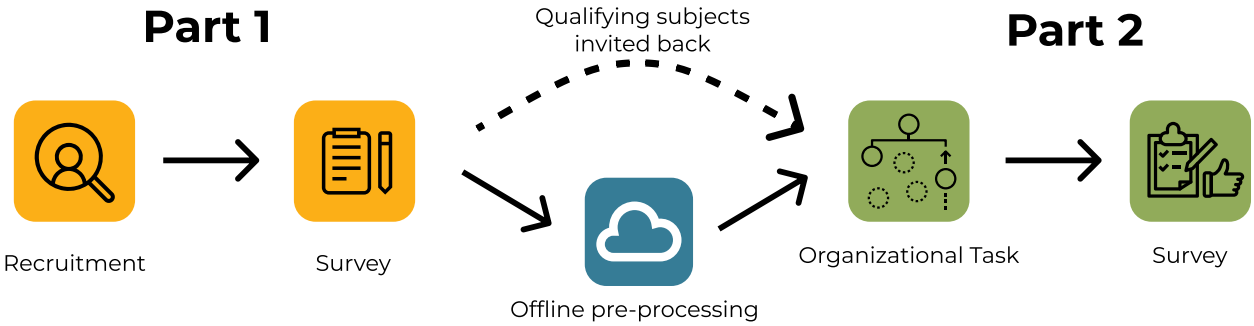


Figure 4.2: Both the Observation Study and Evaluation Study were conducted in two phases to enable offline processing.

scale) with the statement that they would have performed the recommended action anyway (without the recommendation) for two-thirds of the recommendations they accepted. Furthermore, participants who were not shown recommendations independently performed nearly one-third of the actions that would have been recommended. Notably, participants found 15% of the recommendations they accepted surprising, indicating they would not have performed those actions without the recommendation. For nearly three-quarters of accepted recommendations, participants felt the recommendations made organizing more efficient. Our results also suggest future directions for clustering and prioritizing recommendations.

4.2 Observation Study and Evaluation Study Methodology

In this chapter, we conduct two online user studies, the Observation Study and the Evaluation Study. The studies followed similar protocols (see Figure 4.2), so we describe them

together, highlighting key differences. Both studies centered on Google Drive users' own personal cloud repositories, which we accessed using the Google Drive API. We chose to study organization in personal cloud repositories, as opposed to in any other personal file collection, because cloud repositories tend to be smaller and easier to analyze automatically than local storage Capra and Perez-Quinones [2006], the robust Google Drive API enables more privacy-preserving data collection than building our own infrastructure from scratch, and the organization of local storage can be confounded due to operating-system-specific factors Dinneen and Frissen [2020].

We conducted each study in two parts. In Part 1, we recruited participants, had them complete a survey on their usage of their cloud repository, and asked them to grant our code permission to access their Google Drive repository via an OAuth flow. For these and subsequent studies described in this dissertation, in contrast to Chapter 3, we did not collect data for participants on Dropbox. This was because of the small number of participants who used Dropbox in the Investigation Study. Our code subsequently began extracting ten types of file metadata and content features (see Table 3.2 in Chapter 3) for pairs of files in their repository. In Part 2, we invited back eligible participants and asked them to organize their Google Drive repository (see below) and complete a survey that asked about specific actions they did or did not take. The protocols for the Observation Study and the Evaluation Study were the same except for the interface provided for the organizational task in Part 2, as well as the specific survey questions asked in Part 2. The participant pools for the two studies did not overlap.

4.2.1 Recruitment and Part 1

We recruited participants on the Prolific crowdsourcing marketplace (Prolific [2019]). We required participants be age 18+, live in the USA or UK, and have completed 10+ tasks on Prolific with 95% approval. We also required that participants have a Google Drive

repository that was at least three months old and contained at least 100 files.

Once participants had consented to the research, we directed them to grant our code access to their Google Drive repository using the OAuth2 protocol. We used the Google Drive API to analyze their repository, collecting file metadata (e.g., file name, file size), file contents, and Google Drive activity history. In order to protect participant privacy, we did not store the raw file contents. We did, however, extract TF-IDF keywords from files, objects recognized in images using a standard ResNet50 model He et al. [2016], and the names of columns in spreadsheets. We further computed the 10 metadata and content similarity features described in Table 3.2 pairwise between files. Because pairwise comparisons are a quadratic process, for repositories containing more than 1,000 files, we randomly sampled 1,000 files. We additionally collected metadata about participants' past file-management activities in Google Drive's activity log, including what types of actions were applied, the timestamps for those actions, and the IDs of files and folders involved. The purpose was to identify pairs of files that had been managed similarly in the past, which was one factor we used to select file pairs for Part 2. In contrast to Chapter 3, we did not hash file data after processing. This was because doing so would make the data unusable in Part 2, as participants would not be able to see assigned folder and filenames.

In the short Part 1 Survey that followed, we asked general questions about participants' demographics and their use of Google Drive, including their organizational strategies and whether they considered their repository well-organized. Part 1 took 15 minutes on average. Compensation was $2.50.

4.2.2 Part 2

If participants met the eligibility criteria regarding the age and contents of their Google Drive repository, which could only be verified after Part 1, they were invited back for Part 2. We asked them to spend 30 minutes organizing their Google Drive repository using an interface

we provided. We clarified that this interface was a simulated version of their repository, and we emphasized that none of the actions they took would affect their actual Google Drive repository. We further specified that organization could consist of moving files, deleting files, creating folders, and renaming files.

The interface participants used to organize their repository varied across studies and conditions. In the Observation Study, we provided a file browser based on the open-source library Elfinder (Studio 42 [2019]). We chose this interface because it captures many elements (menus, visual design) typical of widely used file browsers. We forked elFinder's code, integrating it with the Google Drive API. For the Evaluation Study, we wanted to design an interface that could integrate recommendations more naturally than the basic elFinder interface. Therefore, we created our own file-browser interface. This interface is shown in Figure 4.7 in Section 4.4. While all participants used this interface, we assigned them uniformly at random to see either a *With Recommendations* or *No Recommendations* variant to let us gauge the impact of recommendations. By random chance, substantially more than half of participants were assigned to the With Recommendations condition. Before beginning the organization task, participants completed a short tutorial highlighting the location of interface components. We required participants to spend 30 minutes organizing.

Participants in both studies then completed a task to characterize the actions they had taken. We showed participants a list of the actions they had performed and asked them to cluster actions into high-level tasks. Participants labeled clusters with free-text descriptions (e.g., "organizing my vacation pictures").

Finally, participants answered survey questions that differed between studies, as well as between conditions in the Evaluation Study. In the Observation Study and in the No Recommendations condition of the Evaluation Study, we primarily aimed to collect data to train and improve our classifier. Thus, participants answered questions about specific pairs of files they had organized in similar ways, or that they had not organized in similar ways

even though our classifier predicted they might do so. If participants moved two files to the same folder, we considered these files to have been managed similarly via move actions. If two files were both deleted, we considered them to have been managed similarly via delete actions. In the Observation Study, our predictions used a rudimentary classifier trained on the data collected from the Investigation Study. In Evaluation Study, we used the classifier as described in Section 4.4. We asked about 14 file pairs as follows, using random pairs of files whenever not enough pairs in the participant's history matched a given criterion:

- 6 file pairs managed similarly, or predicted to be managed similarly, via move actions in our study's organizational task (2 true positives, 1 true negative, 2 false positives, 1 false negative)
- 4 file pairs managed similarly, or predicted to be managed similarly, via delete actions in our study's organizational task (1 true positive, 1 true negative, 1 false positive, and 1 false negative)
- 4 file pairs managed similarly via move actions in a participant's Google Drive activity history (1 true positive, 1 true negative, 1 false positive, 1 false negative)

For participants in the With Recommendations condition of the Evaluation Study, we instead asked participants about the recommendations they were shown. We were interested in participants' reactions to KondoCloud's recommendations, specifically based on what was being recommended (moving or deleting a file) and whether or not the file that spawned the recommendation and the (similar) file for which an action was being recommended were in the same directory. Thus, we asked about up to 15 recommendations shown during the study, selected as follows:

- 4 accepted move recommendations (2 from different folders, 2 from the same folders)
- 2 accepted delete recommendations (1 from different folders, 1 from the same folder)

- 6 rejected move recommendations (3 from different folders, 3 from the same folders)
- 3 rejected delete recommendations (1 from different folders, 2 from the same folders)

In the Evaluation Study, we also asked participants in both conditions additional questions about KondoCloud's interface, including administering the System Usability Scale (Usability.gov [2021]). Compensation for Part 2 of the Observation Study was $10.00. Compensation for Part 2 of the Evaluation Study was instead $15.00 due to the additional time required.

4.2.3 Limitations

Like most user studies, our study is limited by a few factors. Crowdworkers, as a convenience sample, do not represent a broader population. In particular, despite our efforts to protect participant privacy, privacy-conscious crowdworkers were probably less likely to volunteer, potentially biasing the distribution of actions performed. Additionally, our task of having participants organize their Google Drive repository for 30 minutes does not necessarily represent participants' typical behaviors, but rather an idealized scenario. Results about the effectiveness of KondoCloud on such a task therefore may not fully generalize to practice. Participants' self-reported perceptions also may not indicate behavior that would manifest outside of this particular task. Further, our focus on file organization in cloud storage likely does not generalize to other settings, such as local file storage. The typical types of files and typical use cases likely differ between cloud storage and local storage, and some of the features we used (e.g., file sharing settings) are relatively unique to cloud storage (Voida et al. [2013]). Despite our best efforts to provide an interface with minimal confounds, some of our results may be due to idiosyncrasies of the interface (e.g., different right-click menu options) that do not generalize. Finally, although we made a best effort to communicate to participants that no files (including shared ones) would be modified in the course of the study, this

may have caused participants to deviate from their typical file management behavior, either performing more or fewer actions of certain types.

4.3 Observation Study

Our goal for the Observation Study was to characterize strategies for organizing cloud repositories, thereby informing KondoCloud's design. To our knowledge, this is the first study to examine, quantitatively and empirically, users' approaches and strategies when retrospectively organizing the data accumulated in their own Google Drive repository. In contrast, prior studies have examined snapshots of user file collections (outside the cloud context) over time (Czerwinski et al. [2004], Dinneen et al. [2019], Boardman and Sasse [2004]) or asked users to describe, abstractly and qualitatively, how they organize (Oh and Belkin [2014], Oh [2017], Barreau [1995a], Malone [1983]). However, repositories typically do not become more organized over time, and qualitative studies of organization may miss real-time strategies.

4.3.1 Demographics and Cloud Storage Usage

We had 69 participants, 35 women and 34 men. Participants' ages skewed young: 29 were 18-24 years old, 27 were 25-34, and 12 were 35- 64, with 1 who declined to answer. Due to our eligibility criteria, all participants used Google Drive. In addition, 38 used Microsoft OneDrive, 33 used Dropbox, 26 used iCloud, 7 used Sharepoint, and 2 used Box. Participants reported accessing their Google Drive weekly (27), monthly (20), daily (18), or yearly (2); 2 preferred not to answer. Participants interacted, non-exclusively, with Google Drive via the website interface (60), the mobile app (41), and directly synchronizing folders on their computer (22).

Table 4.1 quantifies key characteristics of participants' repositories. Participants had a median of 417 files, a mean of 1,518 files, and a maximum of 12,799 files. Seeing a small number of "power users" with a particularly large number of files is consistent both

	Min	Q1	Median	Q3	Max
# Files	104	228	417	1,448	12,799
(# images)	1	27	114	711	12,030
(# text)	1	13	54	304	7,123
(# media files)	1	8	41	159	1,601
(# spreadsheets)	1	2	3	10	70
(# presentations)	1	1	2	17	1,060
(# other files)	5	47	103	239	2,919
# Folders	2	12	43	100	949
# Avg Files Per Folder	2	8	12	39	134

Table 4.1: Characteristics of participants' Google Drive repositories prior to organization.

with prior work on local file systems (Dinneen et al. [2019]) as well as our findings from the Investigation Study (Chapter 3). Images were the most common file type, with "jpg" (37,349) and "png" (9,768) as the most common file extensions. Text files were the next most common, particularly "pdf" (7,195) and "txt" (7,150) extensions. The "other" category contained a large number of files that either had no extension (1,718) or had a particular user's idiosyncratic file extension (e.g., one participant had 1,473 files for the video game Minecraft). Before organizing, participants had a median of 43 folders, a mean of 118 folders, and a maximum of 949 folders. These observed variations in repository structure were consistent with prior work (Malone [1983], Boardman and Sasse [2004], Hardof-Jaffe et al. [2009], Henderson and Srinivasan [2009], Vitale et al. [2018], Oh [2017]) and our own results in Chapter 3. Via K-means clustering, we found that 47 participants (68.1%) seemed to follow the piler approach, while 22 (31.9%) seemed to follow the filer approach. The cluster centroid for the former was 15.5 folders (with maximum folder depth of 2.7), while the cluster centroid for the latter was 181.3 folders (with maximum folder depth of 8.2).

Participants reported following a variety of organizational strategies in their typical Google Drive usage. Among participants, 28 (40.6%) reported organizing their repository piecemeal when performing other activities, 15 (21.7%) reported organizing their repository across multiple sessions dedicated solely to organizing, and 8 (11.6%) reported organizing

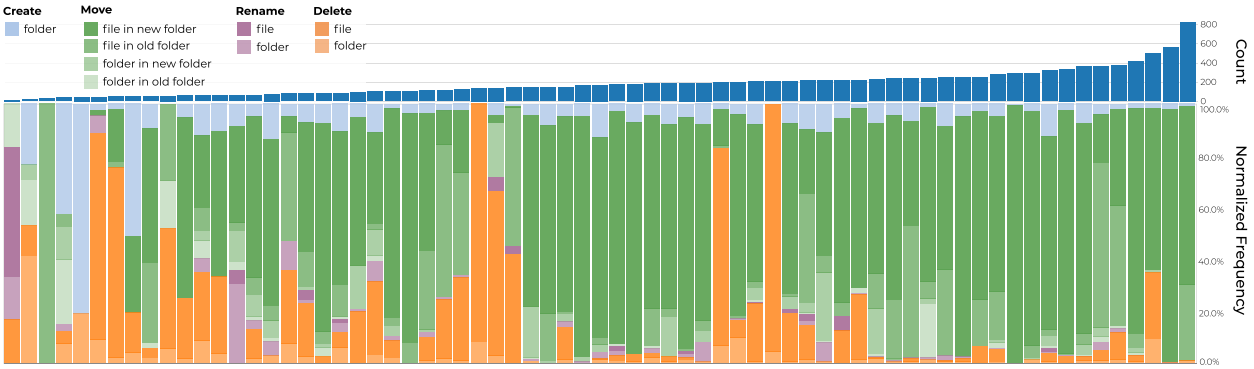


Figure 4.3: Participants' file-management actions while organizing their Google Drive repository in the Observation Study. The x-axis is ordered by the total number of file-management actions the participant took, which is also shown in the bar graph (top). The stacked-bar graph (bottom) shows the distribution of different types of actions.

their whole repository in a single sitting dedicated to organizing. In contrast, 16 (23.1%) reported that they did not organize their repository at all. The remaining 2 participants described organizing files by placing them in the appropriate folders when first saving them, rather than retrospectively.

4.3.2 Strategies in Organizing Repositories

During the organizational task, participants took a total 5,005 file-management actions, including moving, deleting, and renaming files and folders, as well as creating new folders. Of the 5,005 actions, 3,314 (66.2%) were moves, 832 (16.6%) were deletions, 654 (13.1%) were folder creations, and 205 (4.1%) were renames.

Participants varied in the number and types of actions they took, as well as in their organizational strategies. Some participants performed far more actions than others; one participant performed only 12 actions, while another performed 240 actions. The mean number of actions per participant was 72.5, with a standard deviation of 45.7. Figure 4.3 graphs the number and types of actions different participants took, ordering left-to-right by the number of actions taken. It also distinguishes between sub-categories of action types, such as the distinction between moving a file to an existing folder, versus one created during

the study. If a participant acted upon multiple files at once (e.g., highlighting five files and then hitting delete), these are reported separately in this figure. We revisit bulk actions later in this section.

As highlighted in Figure 4.3, participants took very different approaches from each other in the actions they took while organizing their Google Drive repository. The most common organizational strategy was moving files into newly created folders. Notably, 40 participants (58.0%) used this as their dominant strategy. Next most common was a tie between moving files into existing folders and deleting files; each was the dominant strategy for 9 participants (13.0%). The remaining participants used a mix of actions. Most dramatically, one participant only moved files and folders in their 30 minutes of organizing, whereas two others only deleted files and folders. The relative prevalence of different actions was not correlated with the overall number of actions performed.

Participants almost entirely moved files and folders in ways that increased the depth and complexity of their file hierarchies. Figure 4.4 shows changes in the depth (number of parent directories) of files moved during organization. Of the 7,995 files that were moved directly (i.e., excluding files moved as part of moving a folder), 7,797 (97.5%) ended in a directory deeper in the file hierarchy. 5,924 (74.1%) were moved one level deeper. Notably, 4,895 files (61.2%) began in the root directory and were moved one level deeper. On average, file and folder move actions placed items at a file hierarchy depth 1.3 greater (i.e., one folder deeper). Since files in the root directory may represent uncategorized files, a large number of file moves seemed to take uncategorized files and place them in an appropriate folder.

Participants could move or delete a single file at a time, or they could highlight multiple files. This distinction had design implications for the degree to which KondoCloud might consider recommending groups of files to move or delete, as opposed to individual files. Participants performed 2,519 move actions (76.0%) on individual files or folders, and 795 (24.0%) on multiple files or folders. Move actions on multiple files or folders moved a mean

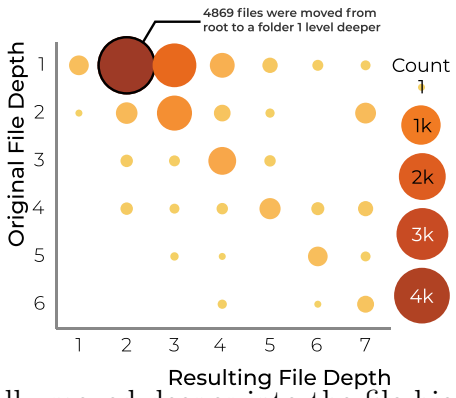


Figure 4.4: Files were typically moved deeper into the file hierarchy regardless of file type.

of 9.4 files or folders at once, with a standard deviation of 16.9 and a maximum of 243. This figure includes moves of the same file or folder multiple times (e.g., a participant could move a file from the root to the "vacation pictures" folder, and then to the "Sardinia" subfolder). Participants performed 728 delete actions (87.5%) on single files or folders, and 104 (12.5%) on multiple files or folders. Delete actions on multiple files or folders deleted a mean of 9.1 files or folders at once, with a standard deviation of 12.5 and a maximum of 70.

4.3.3 File Organization Habits

While participants employed different organization strategies, the specific ways in which they carried out this organization were less variable. Two observations impact design directions: (i) the degree to which participants grouped particular types of actions together, and (ii) the lack of any consistent ordering to groupings of task types. Figure 4.5 shows the relative frequency of pairs of file actions. Three patterns are evident. Participants often grouped folder navigation ("open") actions together, participants often grouped file or folder move actions together, and participants often followed the creation of a new folder by moving at least one file or folder into it. The grouping of move actions resembles Bao and Dietterich [2011]'s idea of task-based context, in which users perform several actions geared toward the same contextual task before switching to another task. This observation suggests that file-management tools should consider task context. KondoCloud's straightforward recom-

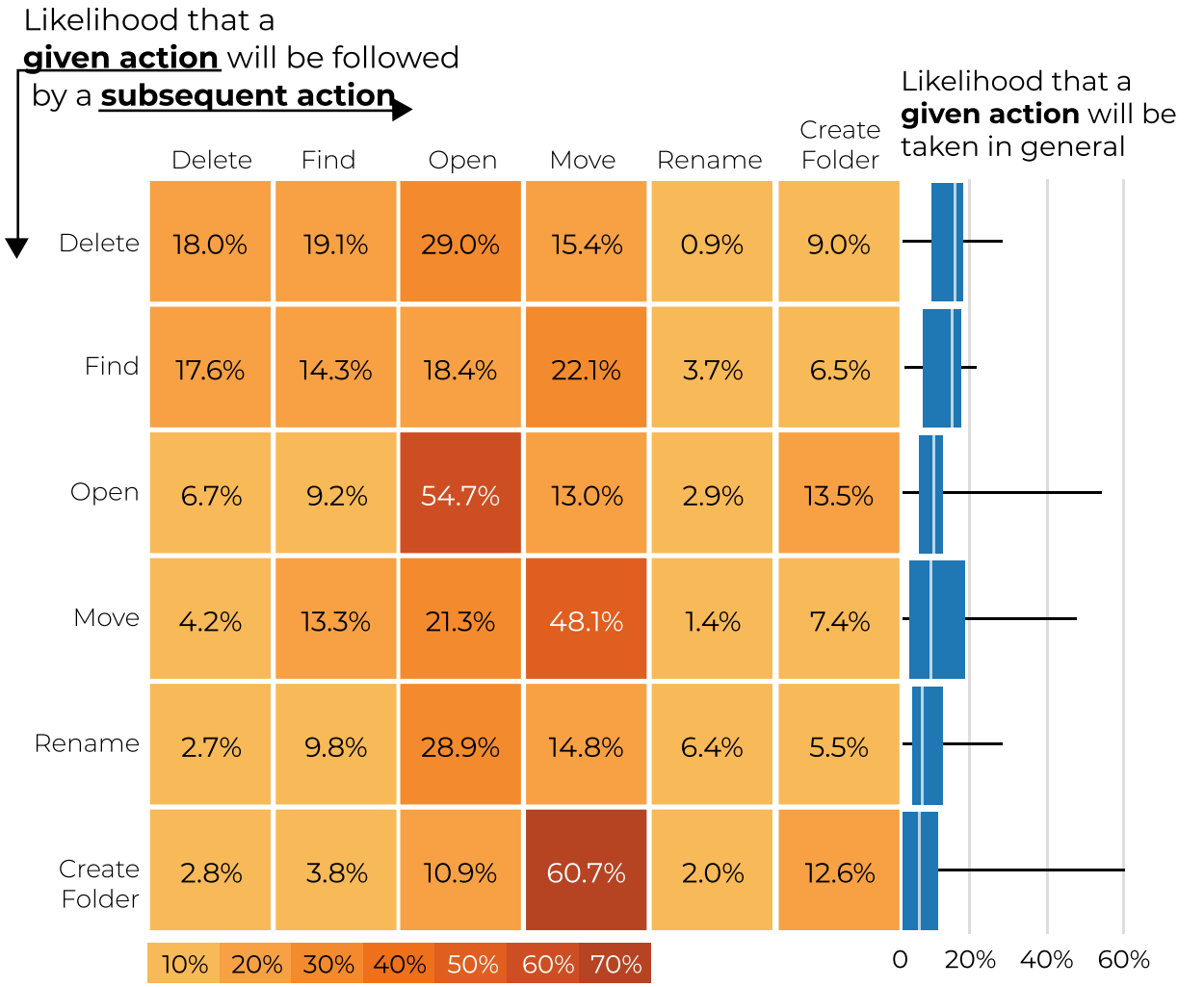


Figure 4.5: Probability of actions following others. Participants often followed moving or opening files with other moves. They also often followed folder creation with moves.

mendations for individual file actions did not capture the observation that file creation events were typically followed by moving files or folders into that newly created folder. We also do not address this in the remainder of the dissertation– this is potentially an open opportunity for future work.

Finally, while we hypothesized that some types of actions (e.g., folder creation actions) might be far more common at certain points during the organization process, we did not find this to be the case. Figure 4.6 shows the number of file actions performed during different temporal segments of the organization task. The mean number of move and delete actions

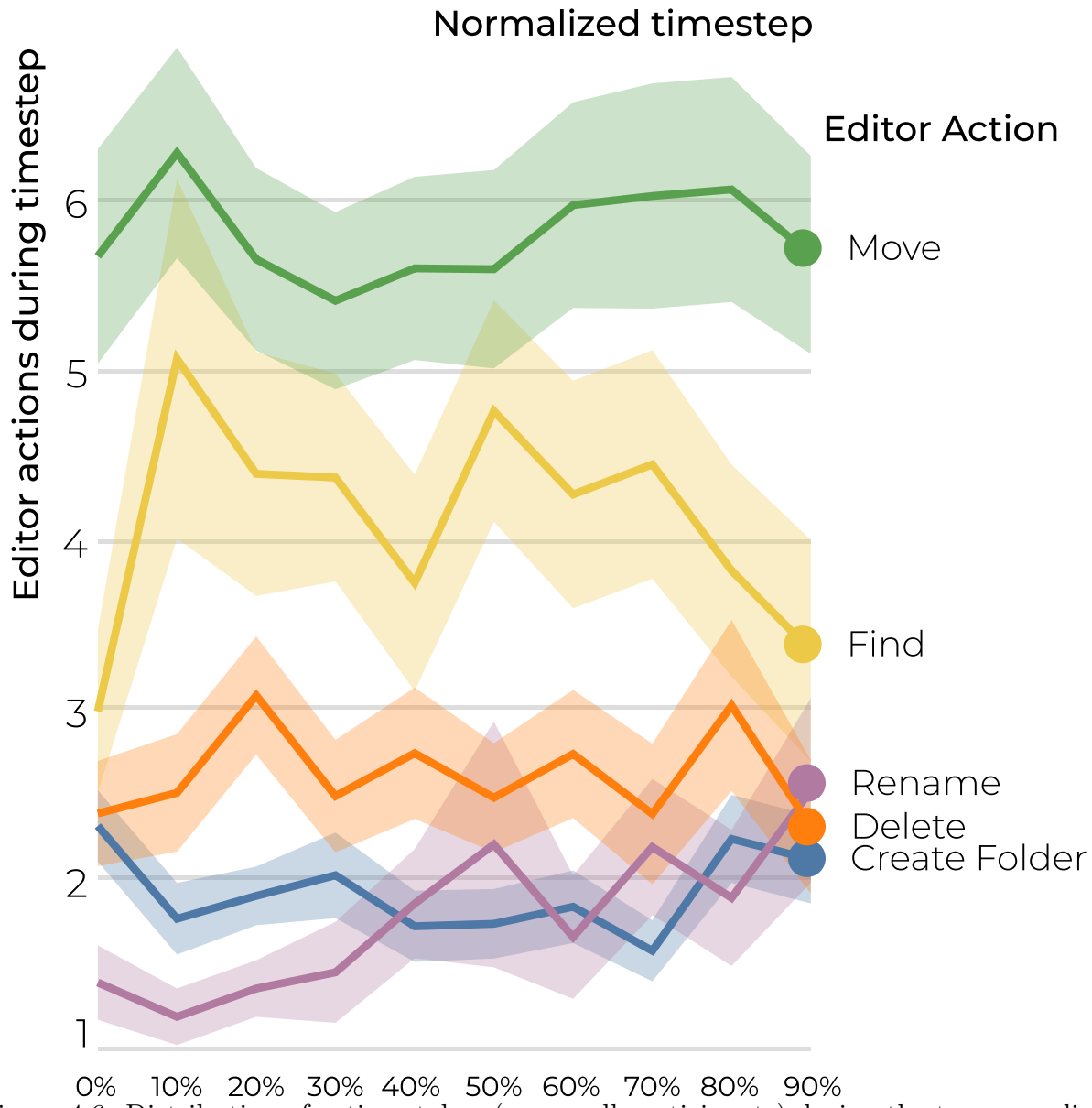


Figure 4.6: Distribution of actions taken (across all participants) during the ten normalized time steps.

during any normalized time block did not differ substantially. This finding suggested a balanced approach for generating recommendations with tools like KondoCloud. That is, the likelihood of recommending particular types of actions should likely not change over time. Tools could perhaps use a particular user's avoidance of certain types of actions early in an organization session to learn to de-prioritize such recommendations.

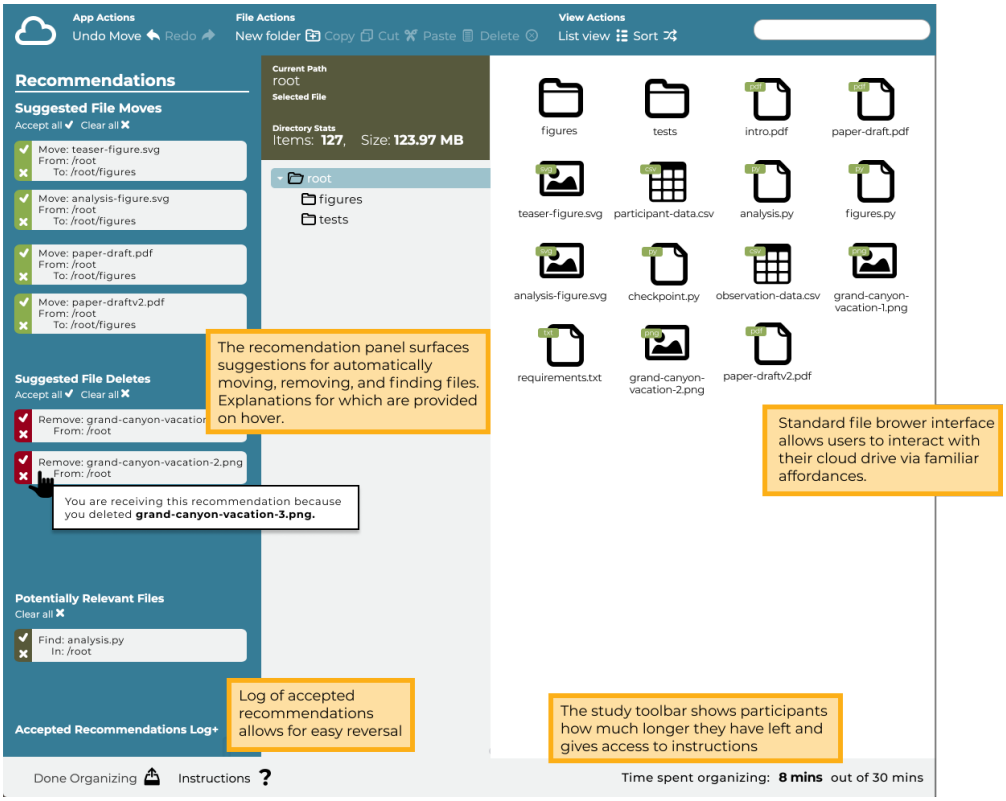


Figure 4.7: The KondoCloud interface augments a traditional file browser with context-dependent, ML-based recommendations.

4.4 The Design of KondoCloud

In this section, we describe and justify the design of KondoCloud, an enhanced file-browser interface that recommends files the user may want to move or delete based on the user having previously taken those actions on similar files. Existing tools offer recommendations for file retrieval or help prevent future disorganization (Liu et al. [2018], Tata et al. [2017], Dumais et al. [2003], Fitchett et al. [2014], Bergman et al. [2019], Segal and Kephart [1999], Sinha and Basu [2012]), but KondoCloud is the first to offer recommendations that retrospectively address existing disorganization in cloud repositories.

4.4.1 Interface Design

The basis for the KondoCloud interface is a standard file browser, shown on the right side of Figure 4.7. Starting from an open source file-browser interface, elFinder (Studio 42 [2019]), we removed unnecessary functionality (e.g., FTP support) and added features offered by common cloud storage browsers (e.g., Google Drive), such as the ability to search by date ranges. We also substantially updated the interface styling to match modern web applications. Using a familiar, non-adaptive, visual basis for the interface was an important design choice because prior work has shown participants have difficulty navigating when files are left "placeless" (Civan et al. [2008], Benn et al. [2015]). This file browser component was the only part of the interface shown to participants in the Evaluation Study's No Recommendations condition.

The second component of the KondoCloud interface is our key novel contribution, the recommendation pane, shown on the left side of Figure 4.7. This component consists of four sub-panes, three of which are always visible, and the last of which can be expanded in an accordion fashion. The first three sections are containers for file move, delete, and retrieval recommendations, respectively. Each recommendation is displayed on a card that contains relevant context for the recommendation. For example, file move recommendations display the file name, where it is currently located, and where the file would be moved to. Clicking on the relevant file or folder names on the recommendation card navigates participants to those files or folders in the main file browser component. We included this ability because prior work suggests that users are unwilling to modify a file location without being able to visualize the spatial movement of the file (Benn et al. [2015]). Hovering over a recommendation card explains the recommendation by showing the file action that triggered the recommendation, as in prior work (Xu et al. [2020]). This format is the same as the *List of Files* summaries that we refer to in Chapter 5. Participants can explicitly accept or reject a recommendation by clicking the respective buttons on the card. All recommendations of a given type can

be accepted or rejected by clicking the button at the top of the recommendation pane. The fourth sub-pane shows the Accepted Recommendations Log and provides shortcuts to reverse accepted recommendations. Following the standard "split" interface model (Sears and Shneiderman [1994], Gajos et al. [2006], Liu et al. [2018]), all functionality offered in the recommendations pane can be performed manually in the standard file browser. We chose this design because previous studies found that moving affordances, instead of copying them into an adaptive component, negatively impacted user satisfaction (Gajos et al. [2006]). We analyze in the Evaluation Study the degree to which participants directly accepted recommendations, versus performing the recommended actions manually in the file browser.

4.4.2 Recommendations

KondoCloud generates recommendations as follows. Each time the user moves, deletes, or previews a file, we use our machine-learning classifier (see Section 4.4.3) to identify similar files. While this initial version of our classifier models only file similarity in making recommendations, future versions could model additional context. Recommendations offer shortcuts to several functions, enhancing a user's capability without removing agency (Heer [2019]). We hypothesized that providing a shortcut to perform the action would enhance file organization in a number of ways (see Section 4.5). More precisely, we choose whether to show a recommendation using a probability threshold as shown in Algorithms 1–2:

These algorithms capture how KondoCloud's recommendations integrate several principles synthesized from our Observation Study. First, the frequency at which recommendations are offered changes in response to how likely a participant is to accept a recommendation of that type. In Section 4.3, we discussed the variance in participants' organizational strategies, particularly in the relative frequencies of the types of actions taken. While we set the same default for every participant based on our classifier training (described below), we updated the decision threshold for our classifier over time for each participant. The default values for

Algorithm 1 AdjustThresholds

```
1: Take arguments initial_change, decay_factor
2: Initialize participant's move_threshold and delete_threshold to default
3: Initialize move_number_of_updates, delete_number_of_updates ← 1
4: while participant organizes do
5:     Participant interacts with recommendations
6:     move_change_value ← CalculateValueChange(
7:   initial_change, decay_factor, move_number_of_updates)
8:     delete_change_value ← CalculateValueChange(
9:   initial_change, decay_factor, delete_number_of_updates)
10:    if participant rejects recommendation then
11:        if recommendation type is move then
12:            move_number_of_updates += 1
13:            move_threshold += move_change_value
14:        else if recommendation type is delete then
15:            delete_number_of_updates += 1
16:            delete_threshold += delete_change_value
17:        end if
18:    else if participant accepts recommendation then
19:        if recommendation type is move then
20:            move_number_of_updates += 1
21:            move_threshold -= move_change_value
22:        else if recommendation type is delete then
23:            delete_number_of_updates += 1
24:            delete_threshold -= delete_change_value
25:        end if
26:    end if
27: end while
```

the variables *initial_change* and *decay_factor* in Algorithm 1 were 0.025 and 0.025, respectively. Because we only displayed recommendations that exceeded the current probability threshold for each action type, accepting recommendations *lowered* the decision threshold for that action type, typically *increasing* the number of recommendations of that type shown. In contrast, rejecting recommendations or letting them expire *raised* the threshold for that action type, typically *decreasing* the number of recommendations shown. We chose for this threshold to decay over time, but not disappear. As discussed in Section 4.3 and Figure 4.6, numerous actions of a particular type could be performed at any point during the study. Correspondingly, even if a user does not accept recommendations of a particular type early on, this does not mean they will not do so later. Algorithm 1 thus ensures that the decision threshold can change substantially even well into the organization process.

Algorithm 2 CalculateValueChange

```
1: procedure CalculateValueChange(initial_change, decay_factor, number_of_updates)
2:     Initialize update, ← initial_change
3:     for i ∈ {1..number_of_updates} do
4:         update ← power(update, 1 + decay_factor)
5:     end for
6:     return update
7: end procedure
```

In keeping with principles identified in prior work, recommendations are easily dismissed or corrected (Amershi et al. [2019]). Further, recommendations are "consistent": only one action is recommended for a file at a time. They are also "polite": after a recommendation is accepted or dismissed, no other action will be recommended for that file for a period of time (Whitworth [2005]). Recommendations are also removed if they are "invalidated," either by a different action being performed on the recommended file or the recommended action becoming impossible (e.g., the file was deleted).

We also designed KondoCloud so that recommendations expire more quickly when file-management actions of a different type are performed. As seen in Figure 4.5, if an action of a particular type is performed, it is more likely to follow or precede another action of the same type than of another type. All recommendations expire after a set number of actions to reduce cognitive load and stay within a user's context (Bao and Dietterich [2011]). However, per Figure 4.5, performing an action of a different type indicates that a user's task context may have changed. Thus, while recommendations expire after any 10 actions (set via pilot testing), actions of a different type count as 2 actions toward expiration. This allows recommendations that are less likely to be accepted to be dismissed more quickly.

4.4.3 Classifier

KondoCloud's recommendations are driven by a set of Logistic Regression classifiers we trained to predict whether two files should be managed similarly. To our knowledge, this is the first classifier for predicting a broad set of file-management actions, such as files to move

and to delete. We trained this classifier based on the 777 file pairs that participants labeled in Part 2 of the Observation Study. In particular, those participants rated their agreement that "these files should be managed in similar ways" for up to 14 pairs of files. As discussed in Section 4.2.2, we intentionally oversampled file pairs that were likely to be managed similarly based on our preliminary notions of file similarity to have more balanced class distribution in training our classifier. We took "strongly agree" and "agree" labels as our positive class, and all other responses as the negative class, creating a binary classification problem. We also examined using only "strongly agree" responses as the positive class, finding it missed cases of interest and suffered from a class imbalance.

We used the ten metadata and content features from Table 3.2 (Chapter 3) as predictive features. Because pairs of files that are both images or both text have additional content features, our overall classifier uses the applicable model among three parallel options (for text-text pairs, image-image pairs, and all other mixed pairs). Considering speed, interpretability, deployment performance, and our small amount of training data by ML standards, we chose logistic regression models. We examined alternative models, including Support Vector Machines, Random Forests, XGBoost, and some ensemble methods. The small improvements we observed in precision were not justified by trade-offs in speed, interpretability, or performance. We used a standard 80-20 train-test split.

Even with our limited amount of training data, our classifier achieved accuracy appropriate for human-in-the-loop recommendations, as shown in our precision-recall curve (Figure 4.8). We achieved F1 scores of at least 0.72 on all three models. While 0.5 is a typical decision threshold, KondoCloud uses the higher starting decision threshold of 0.65 because we focused on providing a smaller number of high-likelihood recommendations, as opposed to many recommendations of potentially lower quality. Spurring this decision, prior work found that a participant's initial sense of an adaptive interface's accuracy influenced later trust in that interface (Gajos et al. [2006], Lee and See [2004]).

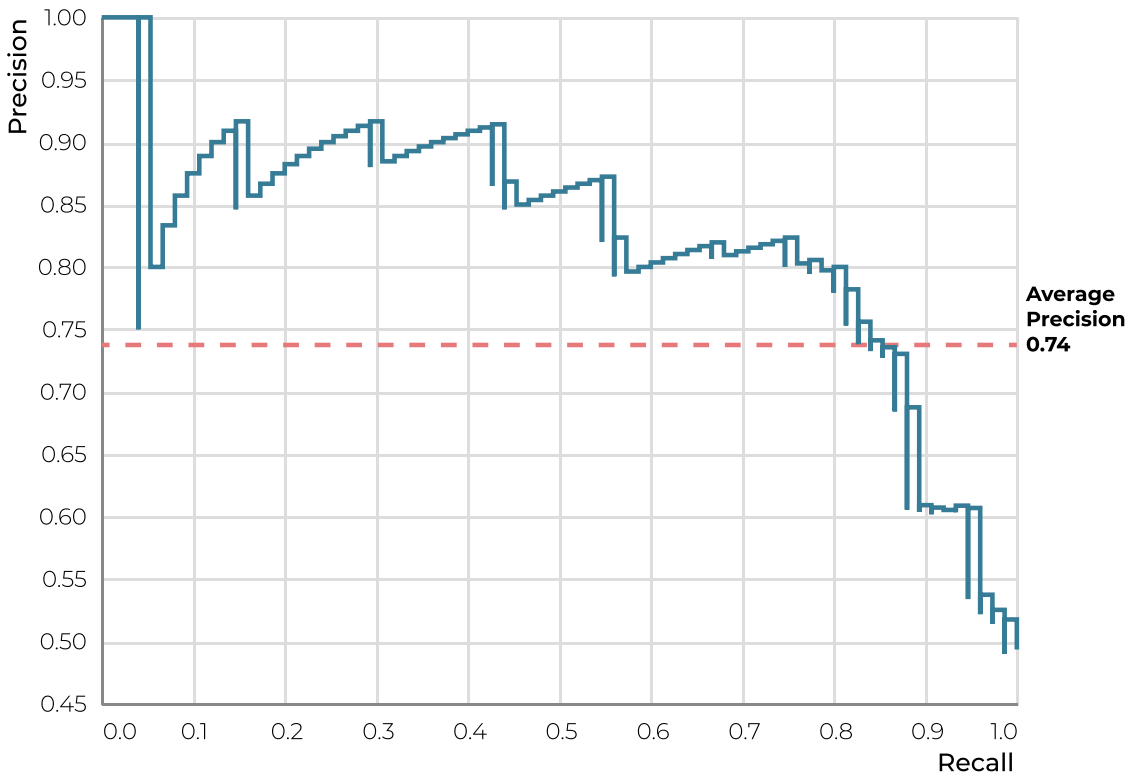


Figure 4.8: Precision-recall curve for the overall classifier.

4.5 Evaluation Study

We evaluated KondoCloud in our between-subjects Evaluation Study. Our key goals were to identify the accuracy and impact of similarity-based file recommendations, as well as to identify ways for future work to improve KondoCloud.

4.5.1 Participants

A total of 59 participants completed the Evaluation Study, 36 in the With Recommendations condition and 23 in the No Recommendations condition. The demographics of the participant population were similar to the Observation Study, with a more even balance among the age of participants. During the study, participants performed a total of 4,644 separate file-management actions, with 3,684 (79.3%) move actions, and 960 (20.7%) deletion actions.

Feature	Mixed pairs	Text pairs	Image pairs
Last Modified	2.884	1.320	2.108
Filename	1.872	0.873	0.557
File Size	0.380	0.955	0.806
Tree Distance	2.163	1.031	1.777
Shared Users	0.668	0.579	0.520
File Contents	1.411	0.102	∼0.000
Text Contents	–	0.319	–
Text Topic	–	0.131	–
Image Contents	–	–	1.008
Image Color	–	–	1.044

Table 4.2: Coefficients (β) of the three Logistic Regression classifiers we created. Our overall classifier (Figure 4.8) chooses the appropriate model based on the types (text, image, or other) of the two files being compared.

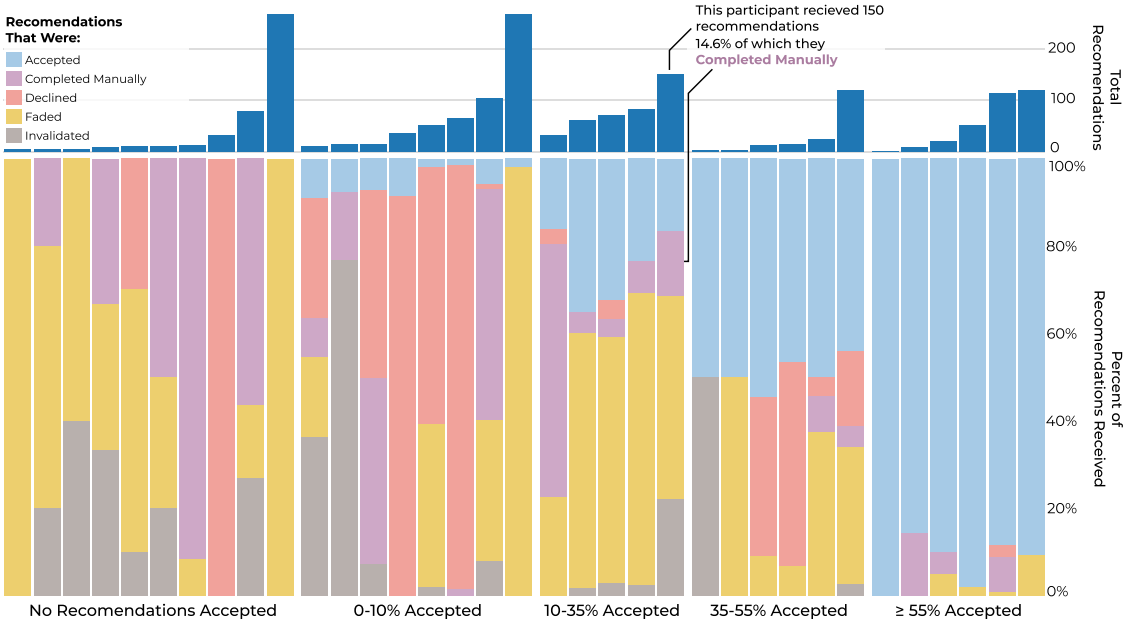


Figure 4.9: The number of recommendations generated for each participant based on their organizational actions (top), as well as the outcome of those recommendations (bottom). We cluster participants on the fraction of recommendations accepted.

Participants again varied in their organizational strategies and actions.

4.5.2 Outcome of Recommendations

KondoCloud's recommendations formed a core component of many participants' organizational workflows. Figure 4.9 shows the number of recommendations offered to each participant and how they interacted with these recommendations. Participants saw 1,856 recommendations: 1,561 (84.1%) move recommendations and 295 (15.9%) deletion recommendations. Participants accepted 473 (25.5%) of these, consisting of 348 move recommendations (22.2% acceptance rate) and 125 delete recommendations (42.4% acceptance rate). In addition, participants manually completed 199 (10.7%) of the recommended actions using the standard file-browser interface while the recommendation was still active. Combining actions taken as a result of formally accepting a recommendation and actions taken manually while that action was also being recommended, 36.2% of recommended actions across participants were completed.

We informally placed the 36 participants in the With Recommendations condition into clusters based on the percentage of recommendations they accepted: participants who accepted $\geq 55\%$ of recommendations (6 participants, 16.7%), 35-55% (6, 16.7%), $10 - 35\%$ (5, 13.9%), 0-10% (8, 22.2%) and those who accepted none (11, 30.6%). As seen in the top portion of Figure 4.9, the number of recommendations generated per participant varied substantially (average of 51.6, standard deviation of 66.1). Because of this, a small number of participants accounted for a large proportion of accepted recommendations. Participants in the $\geq 55\%$ cluster, for example, collectively accepted 282 recommendations, which accounted for 59.6% of total recommendations accepted by all participants. In addition, accepted recommendations made up a significant fraction of the total file-management actions performed by some participants: 14.1% of all move actions and 29.0% of all deletion actions were the result of accepted recommendations.

Recommendations were primarily classified as untaken due to fading away without interaction (i.e., after 10 actions of the same type or 5 of a different type). Of the 1,383 untaken

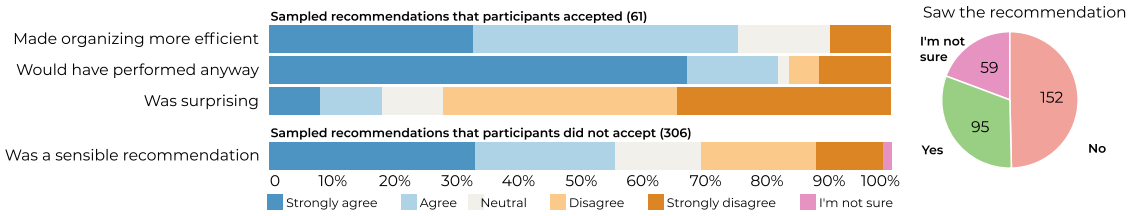


Figure 4.10: Participants' responses to questions about a sample of 61 recommendations they accepted (left, top), 306 recommendations they did not accept (left, bottom), and whether they remembered seeing specific recommendations (right).

recommendations in the With Recommendations condition, 884 (63.9%) faded away, 208 (15.0%) were explicitly dismissed by participants, 199 (14.4%) were completed manually, and 96 (7.0%) were removed due to being invalidated by a participant action. For instance, a move recommendation is invalidated when the destination folder is deleted.

The 25 participants in the No Recommendations condition were not shown recommendations. Nonetheless, for analysis purposes we generated the recommendations they would have been shown had they been in the With Recommendations condition. Participants would have been offered 1,722 recommendations, specifically 1,599 (92.8%) move recommendations and 123 (7.1%) delete recommendations. Though the participants were not shown these recommendations, participants manually performed 32.0% of the move actions and 37.4% of the delete actions that would have been recommended. We did not observe a statistically significant difference between the With Recommendations and No Recommendations conditions in the number of actions performed. Note, however, that our study had a small sample size. Furthermore, the distribution of the number of actions per participant was non-normal, requiring non-parametric tests with lower statistical power.

4.5.3 KondoCloud Usage

Participants in the With Recommendations condition reported several benefits from using recommendations. First, most participants stated that accepting KondoCloud's recommendations improved the efficiency of their organization process. We sampled 61 accepted rec-

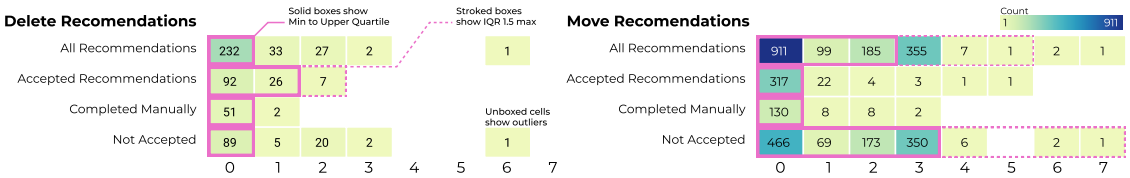


Figure 4.11: Distribution of tree distance between recommended file pairs and the outcome of the recommendation, shown as a heatmap with a boxplot encoded as the borders of the boxes.

ommendations and asked participants to respond to a statement that the recommendation improved the efficiency of organizing their account (Figure 4.10). In 46 (75.4%) of these cases, participants chose "strongly agree" or "agree." Participants also stated that they expected they would have performed the action regardless of the recommendation for 50 (82.0%) sampled recommendations. Some recommendations, however, suggested an action that the participant might not have otherwise taken. In particular, participants responded that the recommendation they accepted was surprising in 11 (18.0%) cases we asked about. Recent work has noted how surprising recommendations can increase a user's satisfaction with a recommender system (Niu and Al-Doulat [2021]). Participants who indicated that a recommendation was surprising offered explanations such as, "How did your system know it was a useless file? Amazed me" and "The two files are not related to each other (to my knowledge) so I was surprised that it made the suggestion." Among recommendations that participants did not accept, many were still potentially desirable. As seen in Figure 4.10, when asked whether a recommendation they did not take was sensible, participants either chose "strongly agree" or "agree" for 170 of the 306 (55.6%) sampled (untaken) recommendations. For 124 of these 170 recommendations (72.9%), participants either indicated that they did not see the recommendation or were not sure whether they had seen it.

We also found that accepting delete recommendations helped participants delete similar files in different folders. We examine this phenomenon in Figure 4.11, which displays the tree distance between similar file pairs for which recommendations were generated. For example, a participant may move an image from the root to the "Vacation Pictures" subfolder, which

generates a recommendation to move another image to that same subfolder. The number of actions needed to navigate from the original image location to the similar file it generated a recommendation for is the x-axis in Figure 4.11. This measure is a proxy for how likely a participant might be to perform the recommendation manually. If the files are in the same directory (tree distance 0), a participant might have already seen the recommended file and already plan to perform the recommended action. If the tree distance is large, however, a participant might not know about the file or otherwise overlook it even though they might wish to manage it similarly to other files. We find that for delete recommendations in the With Recommendations condition, 26.4% of accepted deletion recommendations were for pairs of files in different directories, compared with only 3.8% of recommended deletions performed manually. This represents a significant difference between accepting delete recommendations and performing similar actions manually (Mann-Whitney U test, $p < 0.001$). In the No Recommendations condition, no deletion actions that would have been recommended were manually completed on files in different directories. This suggests that recommendations may have helped participants identify files they wished to delete in different directories. For users of cloud storage who may forget about privacy-sensitive files (Khan et al. [2018]), this form of support could prove useful.

Although KondoCloud generated many move recommendations for similar files at large tree distances, the recommendations that were accepted were typically at much smaller tree distances, as shown in Figure 4.11. This finding is not surprising because most moves are from the root to a subfolder one level below (see Figure 4.4). Such recommendations, where both files are originally in the root directory, would have a tree distance of 0. Indeed, 279 (80.1%) accepted move recommendations moved a file from the root directory to a direct subfolder, and 265 were recommendations that acted on files in the same directory as the originally moved file.

Figure 4.12 shows that 33.3% of participants reported finding recommendations useful,

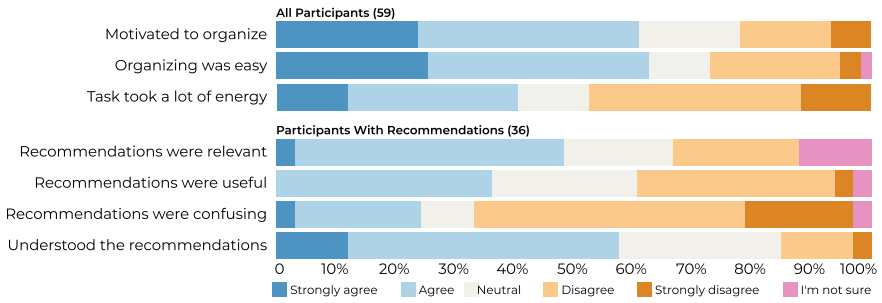


Figure 4.12: Responses to questions about the general organization task (top) and recommendations (bottom).

52.8% reported understanding them, and 44.4% reported finding them relevant. Interestingly, we did not find evidence that these responses correlated with either the number or proportion of recommendations the participant accepted. Participants who did not find recommendations useful reported several reasons why. Some participants simply stated that they would have performed the actions regardless ("Because I would have done it either way"), some did not see them ("Didn't even notice them most of the time"), others preferred to organize manually ("I personally prefer organising files myself rather than trusting suggestions"), and yet others noted that some recommendations could be blocked by others appearing at the same time ("some were useful while some were not and the ones that were not blocked the ones that may have been useful"). Participants across both conditions generally reported being motivated to organize (61.0%), found organizing easy (62.7%), and were roughly evenly split on whether the task took a lot of mental energy (47.5% said it did not). We did not observe a significant difference in the distribution of answers across conditions. Lastly, participants evaluated KondoCloud's usability via the System Usability Scale (SUS). The mean score among participants was 69.9, which is approximately equivalent to the average score in previously evaluated systems (Bangor et al. [2009]). We did not observe significant differences in SUS scores across conditions or relative to the proportion of recommendations a participant accepted.

4.6 Summary

To help users organize their personal cloud repositories, we designed, implemented, and evaluated KondoCloud, a file browser enhanced with ML-based recommendations for moving and deleting files. We conducted two online user studies. In the Observation Study, we observed a variety of organizational approaches, including moving related files to newly created sub-folders, deleting files extensively, and moving misplaced files into existing folders. We also collected data to train a first-of-its-kind classifier that predicts which pairs of files should be managed similarly. In the Evaluation Study, nearly half of participants accepted a non-trivial fraction of KondoCloud's recommendations. A few accepted nearly all. Participants felt recommendations made organizing more efficient, and recommendations for deletion helped participants delete related files located in different directories.

The work described in this chapter is a strong step towards demonstrating feasibility of more complex file management recommendations. However, there were a number of potential improvements identified. The first of these, and the focus of Chapter 5, is that KondoCloud gives recommendations individually, yet many related recommendations may appear in large groups. Participants can scroll to view all of them, but this takes effort. Of the actions on 834 files, 66 (7.9%) produced groups of ten or more recommendations, and one particular action generated 230 distinct recommendations. For these sets of related recommendations, participants typically accepted either most or none of them. Presenting these related recommendations as a group, allowing the user to accept all or reject all, would not only save users time, it would likely improve their understanding of what recommendations are being offered to them. When asked if they wanted recommendations to be shown in groups, 58.3% of participants in the With Recommendations condition responded "strongly agree" or "agree," stating that it would be faster or easier. Consequently, we describe in Chapter 5 a proposed method for summarizing groups of related file management recommendations.

A second improvement would be *modifying our pre-processing approach to improve scala-*

bility would improve KondoCloud's ability to handle large repositories. Because the current pre-processing requires comparing every pair of files, analyzing large file systems is prohibitively expensive. Computing similarity only between a sample of files, as we did, may render some desirable recommendations undiscoverable. Instead, pairwise comparisons could instead be computed at runtime for only the (presumably small) set of files that are moved or deleted and would thus spawn potential recommendations. More advanced techniques could also be applied. Applying locality-sensitive hashing (Gionis et al. [1999]), learned hashing methods (Wang et al. [2015]), or quantization methods (Gray [1984], Jegou et al. [2010], Ge et al. [2013]) could perhaps obviate pairwise comparisons, yet add only mild overhead per recommendation.

Third, *modifying our classifier to model task context* could improve recommendations. Including information like a file's destination might avoid recommending unlikely actions, such as moving a file to a parent folder, as opposed to the more common approach of moving it to a sub-folder. Offering a more diverse set of recommendations to elicit user preferences could also be beneficial (Liu et al. [2010], McCamish et al. [2018], Slivkins [2011]). While our approach of dynamically adjusting the classification threshold based on the user's prior actions personalizes recommendations to some degree, further experimentation is needed.

Finally, *enabling richer interaction with recommendations* could improve usability. For example, Amershi et al. [2014] found that users often wish to give intelligent systems specific feedback, like explaining why an item is labeled incorrectly. Allowing users to identify which file features indicate their personal preference of why files should (or should not be) managed similarly could allow much more targeted and interpretable recommendations (Dasgupta and Sabato [2020]). Further, incorporating early user feedback via more intrusive notifications, such as negotiated-style interruptions (Robertson et al. [2004]) at the start of organizing could enable KondoCloud to personalize recommendations quickly.

CHAPTER 5

FILE RECOMMENDATION SUMMARIZATION

5.1 Overview

The work in this chapter centers on a potential feature to be used in file management recommender systems such as KondoCloud, or systems in prior work (Tata et al. [2017], Xu et al. [2020]). As identified in Section 4.6, particularly in the context of file management recommendations beyond retrieval, large groups of related recommendations may spawn concurrently. When these are presented individually, there is the potential to burden users– typical explanations for recommendations, such as those used in KondoCloud and prior work are insufficient (e.g., "...because you *edited resume2022.docx* on *2022-04-07*", Jahanbakhsh et al. [2020], Xu et al. [2020], Tintarev and Masthoff [2015], Narayanan et al. [2018], Kim et al. [2016]).

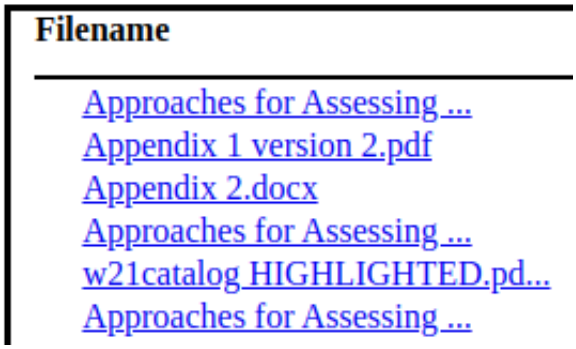
In this chapter, we thus investigate whether related ML-driven recommendations for managing highly similar files in cloud storage can be aggregated effectively. This goal produces challenges related to both the underlying algorithm and the user experience. First, recommendations must be clustered into groups that a user would perceive as actually related, and the algorithm for doing so must be efficient. While future work that builds on this will need to address this in their own manner, we accomplish this step by taking as groups the recommendations generated from a single precipitating action. Second, the system must produce and display a succinct summary of the files included that enables the user to determine accurately which files are being recommended, a task we, and prior work (Narayanan et al. [2018]), term **verification**.

Intuitively, files with similar **characteristics** (e.g., filenames, file extensions, contents, location) that are being recommended for similar reasons are likely candidates for aggregation into a single recommendation that applies to multiple files at once. To this end, we first

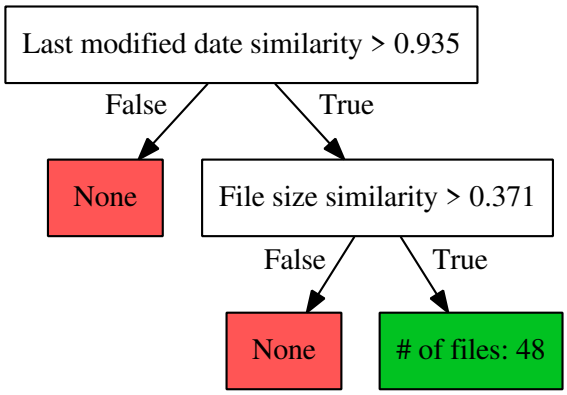
propose an algorithm (Section 5.2) for summarizing related files based on these shared file characteristics. The algorithm takes as input a **group of recommendations**, similar to those generated by KondoCloud for a single precipitating action. As output, the algorithm produces a set of the characteristics shared by all files in the group. While a naive approach would have computational complexity exponential in the space of file characteristics, we develop a greedy approximation algorithm that takes roughly one second on commodity hardware.

The second challenge is to create a representation that helps the user understand which files are included in the group. The most basic approach would be to simply list the files and their most relevant metadata in a table in the user interface. However, this approach is unlikely to scale meaningfully to groups of recommendations that contain many files, and it also does not give any indication about what types of files are excluded from the group. As a result, we develop user-facing **summaries** that leverage our algorithm's output: the shared characteristics of all files in the group (e.g., all documents whose filenames start with 'group-work' and that were modified within a particular date range). We design a text-based summary, termed ***Rules-Text*** and shown in Figure 5.1c, and a visual tree-based summary, termed ***Rules-Tree*** and shown in Figure 5.1d.

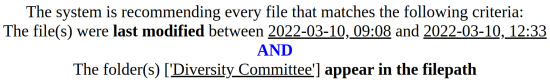
To evaluate our summaries, we conduct a within-subjects online user study, the Explanation Study (Sections 5.3–5.4). We show participants groups of recommendations about their own Google Drive repositories and solicit their perceptions of the associated summaries. We compare the aforementioned *Rules-Text* and *Rules-Tree* summaries we developed with two baselines: simply showing a table listing the files in the group, accompanied by the standard explanation text used in KondoCloud and prior work as described above, termed *List of Files* and shown in Figure 5.1a, and a decision tree, termed *Decision Tree* and shown in Figure 5.1b. We chose the latter since decision trees are often considered among the most interpretable ML classifiers (Lou et al. [2012]).



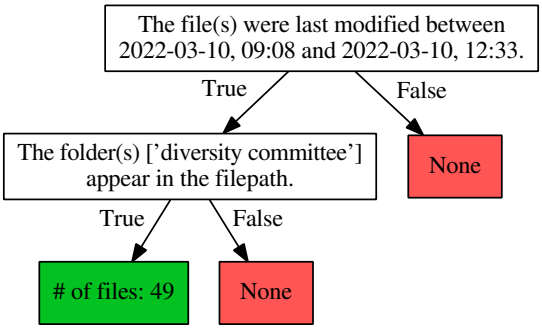
(a) Table listing all files in a group.



(b) *Decision Tree* summary



(c) *Rules-Text* summary



(d) *Rules-Tree* summary

Figure 5.1: To communicate to users which files are contained in a group of recommendations, the most naive approach was to simply list the files (upper left). Our summaries augmented this list with either a decision tree (upper right) as a baseline or the the rule-based summaries we propose in either text-based (lower left) or tree-based (lower right) presentations.

We find that participants perceive our rule-based summaries as less confusing, more helpful, and more verifiable than the two baselines regardless of the number of recommendations in the group. In particular, compared to *List of Files* summaries, we find that *Rules-Text* summaries are 2.7× more likely to have a higher participant rating of helpfulness or verifiability. Further, compared to *List of Files* summaries, *Rules-Text* summaries are 2.0× more likely to have a higher rating of confidence in accepting recommendations without examining the individual files. Contrary to our expectation that participants would prefer visual displays of information, participants rate our text-based summaries slightly better than our tree-based summaries.

5.2 Summarization Algorithm

Here, we describe the motivation for generating summaries, our target format for summaries, and the associated algorithm we created for clustering and summarizing recommendations.

5.2.1 Motivation and Existing Summaries

Summarizing a group of recommendations is necessary to communicate to the user which files are included, and which are excluded from the group. While summaries are useful for file retrieval (viewing a file), they are even more important for destructive and permanent actions like deleting or moving files. This observation is notable since recent research has increasingly focused on tools to help users delete and move files to improve personal information management (Dropbox [2022], Khan et al. [2021], Bergman et al. [2019]). Furthermore, even if multiple recommendations for file retrieval were summarized, the user would likely still view those files individually and sequentially, in contrast to bulk file deletion or bulk file movement.

If multiple recommendations are grouped and summarized in a way that the user trusts to convey which files are included, the user can accept them together, improving efficiency and increasing the user's confidence that related files have not been inadvertently excluded from the recommendation. Our summaries thus aim to empower users to quickly determine which files are covered by a summary, a task we, and prior work (Narayanan et al. [2018]), call *verification*.

We evaluate four summary types: *List of Files*, *Decision Tree*, *Rules-Text*, and *Rules-Tree*. The former two are intended as baselines, whereas the latter two are novel contributions. Summaries for file recommendation in current systems generally follow the form, "You {performed action} to {file name} in {time period}" (Chapter 4, Xu et al. [2020]). We mirrored this phrasing in our *List of Files* baseline, and we also accompanied it (and all other summaries) with a table listing the files in the group, as shown in Figure 5.1a.

We expected these explanations to fall short when recommending that the same action be applied to multiple files. The user might wonder how the files listed in the table relate to each other, or whether other files with similar characteristics were mistakenly left out.

Our second baseline is based on an observation from efforts in interpretable ML. Decision tree classifiers are typically considered among the most intelligible types of ML models (Lou et al. [2012]). In particular, our *Decision Tree* baseline displays a visual tree-based representation of a decision tree classifier that is used to select files for the group of recommendations based on their similarity to a file spawning the recommendations (e.g., deleting Northern-Lights_98.jpg might spawn recommendations to delete other, related files). As shown in Figure 5.1b, the visualization of the decision tree references the kinds of information used by the classifier (e.g., a normalized quantification of the similarity of file names). Despite the inherent interpretability of a decision tree, we expected that the model parameters would prove somewhat unintelligible to non-experts. For a linear model on a bag-of-words featurization, for example, each word is assigned a β coefficient to best match the decision boundary of the original recommendations. Understanding whether a file would be recommended or not would therefore require a complicated calculation for a non-technical user. This is exacerbated when inputs are more heavily featurized, such as in an embedding space (Lipton [2018]). Non-parametric methods such as K-nearest-neighbors are no better, since the distance metric will have the same difficulties as uninterpretable model parameters.

5.2.2 Structure of Rule-based Summaries

Table 5.1 details the format of the rule-based summaries we developed: *Rules-Text* and *Rules-Tree*. These summaries consist of the intersection of multiple predicates on the characteristics of the files in the group (Table 5.2) presented in ways we designed to be interpretable to non-technical users. Intuitively, these predicates represent characteristics of the files included in as group of recommendations. These predicates take two forms depending on the data

Summaries	P	$::=$	$(r \mid s) \wedge ... \wedge (r \mid s)$
Range Predicate	r	$::=$	$n_1 \leq x \leq n_2$
Set Predicate	s	$::=$	$(c_1 \in x) \wedge ... \wedge (c_n \in x) \mid s \vee s$

Table 5.1: The structure of our proposed summaries.

type of the characteristic. For numeric characteristics, such as the file size or last modified date, the predicate covers a range of values (e.g., "files between 3 and 5 megabytes"). For set-based characteristic (all others, such as the set of objects recognized in an image), the predicate evaluates to true if, for at least one of the subsets of items in the predicate, the file's relevant feature set contains all of the given items. For example, if a predicate on filename tokens takes the conjunction of the sets "['course', '2019'] OR ['course', '2020']", then any file with filename tokens containing either subset will be covered by the predicate. To limit the computational cost and ensure simplicity of summaries, we allow no more than a single "OR" conjunction for a particular feature predicate. We also do not allow "OR" clauses between predicates / different features (e.g., "The folder(s) ['work'] appear in the file path OR the filename(s) start with 'budget_' "). Given that we showed the notion of similarity strongly informs desires about richer file management actions in Chapter 3, and given that the displayed predicates can easily explain multiple recommendations at once, they seem to address the expected drawbacks of the baselines above.

Because we were also interested in how the summary was presented to users, we developed and tested two visual presentations for rule-based summaries. The *Rules-Text* summary shows a plaintext representation, as in Table 5.1, with minor embellishments (e.g., bolding) for readability. The *Rules-Tree* summary inserts predicates into the same tree structure used in our *Decision Tree* baseline.

Attribute	Predicate Type	Example
Filename Prefix	Set	The filename(s) start with 'bronze-age'
Filename Tokens	Set	The filename(s) contain sub-part(s) ['group', 'work']
File Extension	Set	The file(s) have the extension 'png'
File Path	Set	The folder(s) ['useful'] appear in the filepath
Shared Users	Set	The file(s) are shared with ['example@gmail.com']
Recognized Objects	Set	The system thought it saw the object(s) ['website', 'letter'] in the image(s)
File Text Tokens	Set	The file data contains the word(s) ['earnings', 'call']
File Size	Range	The file(s) have size from 2.0 Kb to 1.0 Mb
Last Modified Date	Range	The file(s) were last modified between 4/7/2019 14:40 and 4/8/2019 14:45

Table 5.2: The file characteristics used in summaries, their predicate types, and examples of their text representations.

5.2.3 Synthesis Algorithm

Synthesizing summaries in the form of Table 5.1 over multiple recommendations faces several challenges. First, the synthesized summary is highly unlikely to be able to exactly match the group of recommendations output by the original recommender system. This is only a minor concern in prior work, as researchers either tune the neighborhood around a single example to be summarized such that summaries are rarely untruthful (Ribeiro et al. [2016]) or assume a particular model form for the recommender system (Sharma and Cosley [2013], Zhang et al. [2014], McInerney et al. [2018]). We instead modify the set of recommendations included in a group to exactly match those covered by the summary. We feel that doing so is a potentially beneficial form of regularization on the recommender system output. However, it is still desirable to match the original set of recommendations in a group as closely as possible. To do this, we select among explanation candidates using the F_β score, where the positive labels are the original recommendations. We set weights on recall versus precision via pilot testing. Second, finding a globally optimal candidate for set-based predicates may require enumerating an exponential number of candidates in the worst case. To address this, our synthesis algorithm greedily adds tokens to the potential set predicate. This takes time $O(nk)$, where n is the number of possible tokens to explain over, and k is the number of tokens in the optimal predicate. We find that k is usually small (< 5) in practice. In addition, we limit the number of tokens examined per file to 1,000 for our experiments.

Future work may examine the practicality of this limit. Third, to integrate seamlessly with the underlying recommender system, summaries must be generated in close to real time. Thus, we compute an approximation by greedily selecting the best predicate to add to the current set.

With these challenges in mind, we synthesize summaries using Algorithm 3, which takes Algorithm 4 as a subroutine. Informally, Algorithm 3 looks at each attribute, and uses a subroutine to identify the best predicate for that attribute given the current set of items covered by the explanation. Whichever one yields the most improvement in the F_β score is added to the explanation. The algorithm halts when adding a predicate on another attribute would negatively impact the score. The set of files covered by a candidate explanation is identified via pre-built sorted range or reverse-index data structures that enable efficient lookup. The best candidate for set-based explanations is approximated with Algorithm 4, while the best candidate for attributes that take range-based predicates is found by enumerating all choices. Building the data structures and enumerating solution candidates are viable in practice because the universe of constants for range predicates and tokens to be added to set predicates is restricted to values drawn from the original group of recommendations. Intuitively, choosing a value in a range predicate that was not drawn from a recommended item cannot improve more than one drawn from a recommended item and can only negatively impact precision. A similar principle holds for tokens in sets that do not apply to any files in the group. While we find that the given synthesis algorithms are efficient in practice, we do not explore the optimality gap due to approximation, nor do we explore the potential for more efficient implementations.

5.3 Explanation Study Methodology

To study the effects of different summary types, we conducted a two-part, within-subjects online user study (the Explanation Study) with 44 participants. In Part 1, we scanned

Algorithm 3 Full Approximation algorithm

procedure FULLAPPROX(Files in Recommendations, Other Files)
    $RunningBase \leftarrow$ Files in Recommendations
    $RunningOut \leftarrow$ Other files
    $Summary \leftarrow []$
    while Haven't used all attributes **do**
        $Scores \leftarrow []$
        for Attribute in unused attributes **do**
            $Predicate, Score \leftarrow BestPredicate(Attribute)$
            $Scores \leftarrow Scores + [Score]$
        end for
        if Best score ≥ 0 **then**
            $Summary \leftarrow Summary + [Predicate]$
            $RunningBase \leftarrow RunningBase \cap$ (Files in Recommendations covered by predicate)
            $RunningOut \leftarrow RunningOut \cap$ (Other Files covered by predicate)
        else
            break;
        end if
    end while
    return *Summary*
end procedure

Algorithm 4 SetGreedy

procedure SETGREEDY(Attribute, RunningBase, RunningOut)
    $Predicate \leftarrow []$
    $Scores \leftarrow []$
    while Some set elements remain unused **do**
        for *set_element* for Attribute **do**
            $Scores \leftarrow Scores + [ChangeInSummaryScore()]$
        end for
        if Best Score > 0 **then**
            $Predicate \leftarrow Predicate +$ [Best *set_element*]
        else
            break;
        end if
    end while
    return *Predicate*
end procedure

the participants' Google Drive accounts, pre-computed groups of recommendations, and generated summaries of each of the four studied types (*List of Files*, *Decision Tree*, *Rules-Text*, *Rules-Tree*) for every group. We used stratified sampling to select up to 14 group / summary pairs. We presented these to participants in Part 2, asking them to evaluate attributes like the helpfulness and verifiability of each summary.

5.3.1 Part 1

We recruited crowdworkers from the USA and UK through Prolific [2019].We required that participants had completed 10+ submissions with a 95%+ approval rating and had Google Drive accounts that were 3+ months old and contained 100+ files. Once we recruited participants and they had consented to the research, they granted our web application access through OAuth 2 to scan their Google Drive files' data and metadata. Participants were then directed to a survey on their demographics and usage of cloud storage. Part 1 took approximately 15 minutes. Compensation was $5.00.

We pre-computed file recommendations using the method from Chapter 4. As before, we limited computation to all pairs of at most 1,000 files. We generated groups of recommendations by iterating over all files, sequentially designating each as the "base file." All files classified as similar to the base file were recommended as a group. To limit overlap, we did not generate a group for the base file if that file appeared in a previous group.

For each group, we then generated a summary of each type identified in Section 5.2 (*List of Files*, *Decision Tree*, *Rules-Text*, *Rules-Tree*). We excluded the base file from this summary as it was used to generate the "scenarios" described below. As described in Section 5.2, we modified the set of files in a group to exactly match those covered by the summary. The *List of Files* summaries require no generation, the *Rules-Text* and *Rules-Tree* summaries were generated with Algorithm 3 and the *Decision Tree* summaries were generated by training decision trees (Gini impurity, max depth of 2 set in pilot testing) that took the original group

of recommendations as positive labels, and files not recommended as negative labels.

Once summaries were generated, we used stratified sampling to choose group / summary pairs to present in Part 2. We selected up to 14 groups as follows:

- 4 groups, based on summary complexity (2 "complex", 2 "simple")
- 4 groups, based on "discriminativeness" (2 "discriminative", 2 "non-discriminative")
- 6 groups, based on size (2 "small", $\leq$ 25th percentile of group size for participant, 2 "medium", 25th–75th percentile, and 2 "large", $>$ 75th percentile)

We labeled *Rules-Text* or *Rules-Tree* summaries as complex if they required at least one 'AND' or 'OR' keyword, and *Decision Tree* summaries as complex if the resultant tree had depth > 1. *List of Files* summaries were not complex. We identified groups as discriminative based on what percentage of the files in a folder were recommended, among folders that contained recommended files. Intuitively, recommendations that suggest performing an action on all files in a folder (recommendations that are not "discriminative" of files in a folder) are less helpful for users, given that such files can easily be identified by the user themselves. In contrast, selecting a specific subset of files from a folder may require more effort from a user, and such recommendations are therefore more helpful. If there were fewer group and summary pairs that met the complex and discriminative criteria than desired, additional summaries were sampled from the small, medium, and large groupings.

5.3.2 Part 2

We invited back eligible participants after we had finished the processing of Part 1. When participants returned, they were presented with instructions that explained we would show them up to 14 hypothetical "scenarios" (the **Scenario**), based on a group / summary pair, each of which read, "Suppose that you shared, moved, or deleted {base file}". We presented the group of recommendations (**Recommended Files**) in a table with relevant metadata

that linked to the file data in Google Drive (Figure 5.1a), along with the summary (the **Explanation**). For *List of Files* summaries, we presented only the text, "Because you shared, moved, or deleted {base file} ({file path of base file})". Other summary types were displayed as in Figure 5.1. The visual summary types, *Decision Tree* and *Rules-Tree*, also had a hover interaction on leaf nodes that displayed the names of the files allocated to that node. We then asked participants a set of 8 questions (shown in Table 5.3) about the scenario, group, and summary. After completion, participants were redirected to the link for compensation. Part 2 took approximately 1 hour to complete, and compensation was $15.00.

5.3.3 Limitations

Our study required that participants accept permissions allowing our web application to view and download their file data—privacy-conscious participants may have been unwilling to participate. In addition, our study presents hypothetical scenarios. While this allows us to directly study groups of file recommendations, participants' survey responses may be biased either towards accepting recommendations, because there was no cost to agreeing, or against accepting them, because of the uncertainty introduced by lack of context. Further, although it was necessary from a computational standpoint, limiting our all-pairs similarity to 1000 files may bias our results. The absence of recommendations that would have been included, had the files been sampled for similarity, may negatively bias participants' survey responses for summary types based on pre-computed similarity (*List of Files*, *Decision Tree*). Our study was also conducted on crowdworkers. Prior work has shown that crowdworkers are not representative of any broader population, and that many skew younger and more technically-savvy.

5.4 Results

We describe our participants and their survey responses, then build a set of regression models to identify the effect of summary type on qualities such as understandability, helpfulness, and verifiability.

5.4.1 Participants

We recruited 44 participants for the Explanation Study. 29 (67.4%) participants were female, 11 (25.6%) were male, and 3 (7.0%) were non-binary. Most participants were 25–34 years old (16, 36.4%), with a similar number (15, 34.1%) 18–24 years old, and the remaining, 35–64 years old. Most (35, 79.5%) had no computer science background. Participants interacted with their Google Drive account in various ways. Participants used Google Drive through the website (37) or the mobile app (30) nearly equally, though a few synced folders directly from their local storage (12). Most participants interacted with their account weekly (17, 40.4%), though monthly (13, 31.0%) and daily (11, 26.2%) usage was also common. Participants generally disagreed that their accounts were well-organized (15, 34.1% "disagree", and 13, 29.5%, "strongly disagree"). Participants also generally agreed that their files were "uncategorized" (15, 34.1%, "agree" and 13, 29.5%, "strongly agree").

The distribution of participants' cloud storage files was similar to those from the Investigation Study, the Observation Study, and the Evaluation Study. We processed 97,546 files from participants. The median participant had 1,310.5 files in their account, and the mean participant had 2,217 files, with a standard deviation of 3,622.5. The smallest account had 117 files, and the largest, 16,137 files. Most files were images (43,889), with a large number of media (14,791) and text files (14,199 across participants). Most images were "jpg" files (35,085), most media files were "mp3" (4,164) or "heic" files (4,049), and most text files were "pdf" files (9,790). There was also a long tail of 23,172 files with uncategorized extensions. These included "no extension" (5,131), Autodesk files ("flc", 1,893) and paintbrush bitmap

files ("pcx", 651).

5.4.2 Survey Responses

Participants saw 563 scenarios. Summary types appeared in roughly equal numbers of scenarios: 131 (23.3%) *List of Files* scenarios, 153 (27.2%) *Decision Tree* scenarios, 132 (23.4%) *Rules-Text* scenarios, and 147 (26.1%) *Rules-Tree* scenarios. The sampling reasons were roughly evenly distributed as well. The most common choice was summaries over small file groups (90, 16.0%), and the least common choice was non-discriminative summaries (64, 11.4%). The size of the recommendation groups followed roughly a power-law distribution: the mean sampled group contained 40.2 recommendations, while the median sampled group contained 7 recommendations. The largest group sampled was 1,179 recommendations. On average, groups identified by *Rules-Text* and *Rules-Tree* summaries were larger: groups had median size 9 for both, compared against median sizes of 6 for *List of Files* and *Decision Tree* summaries, respectively. However, per the discussed limitation in Section 5.3.3, this is likely to be biased. Differences between summary types carried over to the scores: *Decision Tree* summaries had an average F_β score of 93.0, while *Rules-Text* and *Rules-Tree* summaries had a score of 68.9. This is a notable difference, but there are several considerations. First, again due to the limitation in Section 5.3.3, scores for *Decision Tree* summaries are biased upward, as they are fitting a smaller set of files. Second, scores only indicate a summary's ability to match the original classifier recommendations. This is independent of participants' perceptions of the recommendations and summaries, which is the focus of our analysis.

The file characteristics (Table 5.2) chosen to summarize over were roughly consistent across summary types, as seen in Figure 5.2. We display only *Decision Tree* and *Rules-Text* summaries, because *List of Files* summaries do not explain over characteristics, and *Rules-Tree* summaries are generated in the same way as *Rules-Text* summaries. Their distributions are essentially identical. By far, summaries most commonly used filenames and last modified

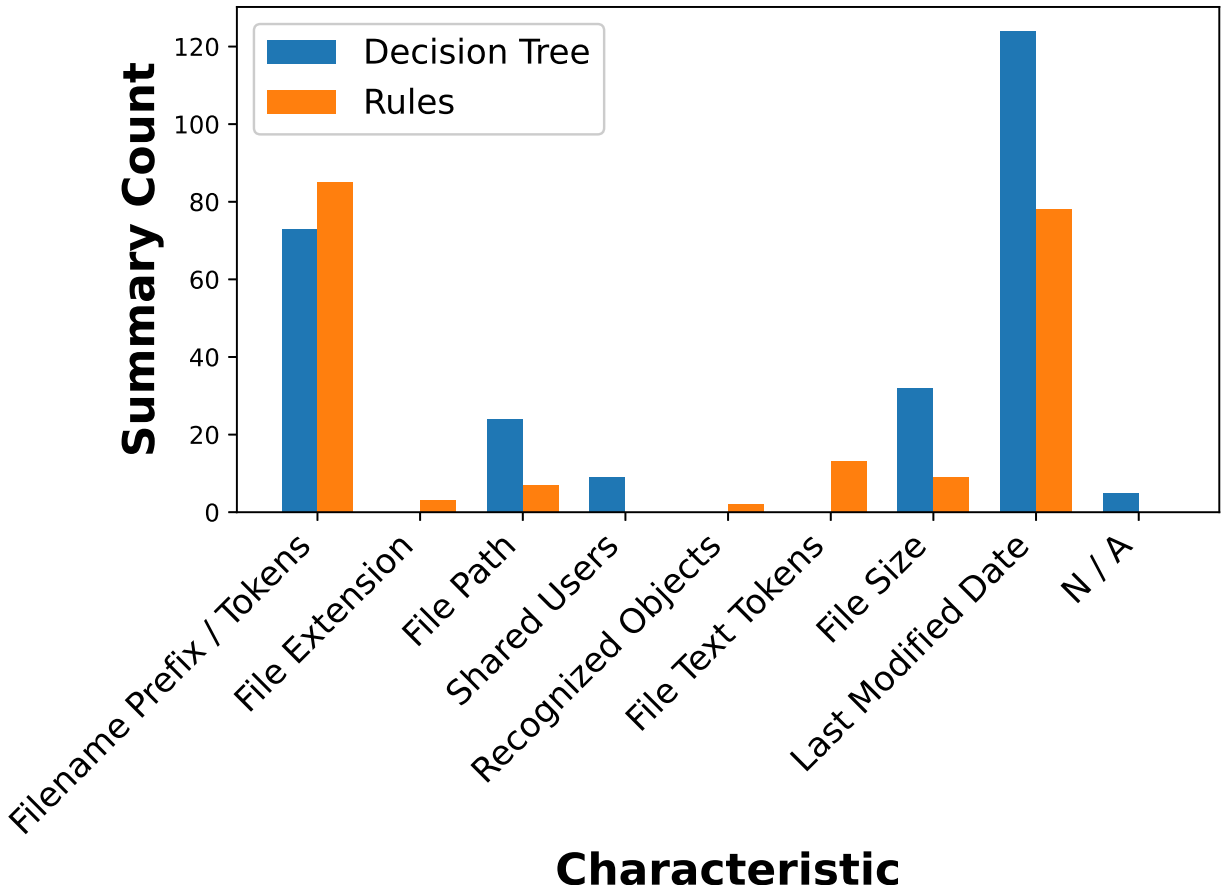


Figure 5.2: Number of times each characteristic appeared in a summary for *Decision Tree*, or a *Rules-Text*/ *Rules-Tree*. "N/A" represents *Decision Tree* features not available for rules.

dates. File path and file size characteristics were occasionally present in both summary types, and the remainder of characteristics were rarely included in summaries. Some characteristics were not present for a particular summary type. Because the original classifier does not have features for *File Extension*, for example, this could not be selected by the *Decision Tree* summary. Similarly, *Rules-Text* summaries did not have predicates for file content, topic, and image color features. In each of these cases, the relevant characteristics were not commonly summarized over, and therefore were not a key differentiating factor between the summary types.

We display the proportion of Likert-scale responses for each question from Table 5.3 in Figure 5.3. We note the difference between two types of questions asked: "Group-Based"

Q1	The **Recommended Files** are related to each other
Q2	I could accurately describe to someone else what the **Explanation** is saying
Q3	The **Explanation** is confusing
Q4	I'd find a style of of explanation similar to this **Explanation** helpful when files are recommended to me
Q5	If I saw a table of all the files in my Google Drive, I could pick out which ones the **Explanation** covered
Q6	Based on the **Explanation** given, I believe the system sees the **Recommended Files** as related for the same reasons I do
Q7	I would perform the same action as in the **Scenario** on the **Recommended Files**
Q8	After seeing the **Explanation**, I would feel more confident performing the same action as in the **Scenario** on the **Recommended Files** without examining every file individually

Table 5.3: Questions shown to participants for each scenario in Part 2. We referred to groups of recommendations as "**Recommended Files**", the summary as the "**Explanation**", and the file action producing the recommendations as the "**Scenario**"

questions, such as Q1 and Q7, could be answered without reference to a summary. The only impact that a summary had for these questions was that it would change the group of recommendations generated. We use "Group-Based" questions both to analyze our recommendations compared to the Investigation Study, the Observation Study, and to control for mostly-summary-independent aspects in our regressions (Table 5.4).

We find that answers to the "Group-Based" questions were roughly in line with expectation from prior work. In Q1, participants generally found the files that were recommended

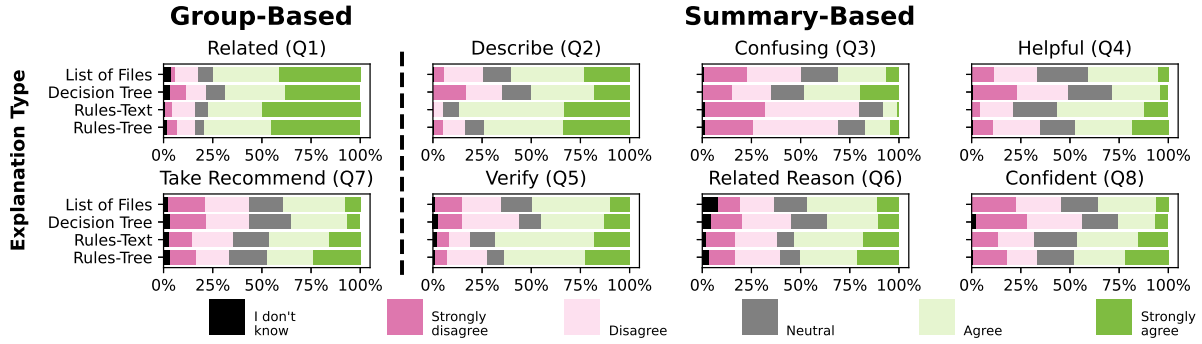


Figure 5.3: Proportion of Likert scale responses to each question, separated by summary type. "Group-Based" questions are those that are answered without reference to a summary, while "Summary-Based" questions referred explicitly to a summary.

to be related ($> 50.0\%$ "agree" or "strongly agree" responses). Given that this population is sampled from items that a classifier identified as likely to be related, this is roughly within expectation (cf. Chapter 3). While there were not significant variations between summary type for Q1, we note that the proportion of "strongly agree" or "agree" responses for *Rules-Text* and *Rules-Tree* summaries were equal to or slightly greater than the other summaries. Given that the only difference between summary types for this question was the set of recommendations, this supports the idea that the regularization induced by our proposed summaries is, at least, not harmful. Future work may investigate this in greater detail. In Q7, participants indicated they would accept the group of recommendations ("strongly agree" + "agree") for between 1/3 and 1/2 of scenarios across summary type. This is comparable with, though slightly higher than, the individual acceptance rates we observed in the Evaluation Study (Chapter 4). It is possible that the presence of summaries increased participants' willingness to accept recommendations, but it is also possible that the hypothetical nature of the scenarios was the reason behind these higher acceptance numbers. Future work examining these summaries in full interfaces will be useful for answering this.

Participants generally found our summaries (*Rules-Text* in particular) more understandable, less confusing, more helpful, and more verifiable than *List of Files* or *Decision Tree* summaries. The responses to Q2 suggest that participants overall could describe each summary type (proportion of "agree" and "strongly agree" responses were $> 50\%$ across summary types). Pilot testing suggested that the concrete task in Q2 was a reasonable proxy for "understandability". Visual inspection indicates that the *Rules-Text* and *Rules-Tree* summaries have a higher proportion of "strongly agree" or "agree" responses than *List of Files* or *Decision Tree* summaries for this question: participants answered "strongly agree" or "agree" in 86.4% of scenarios for *Rules-Text* summaries, and in 74.0% of scenarios for *Rules-Tree* summaries. Roughly similar proportions of responses are seen in Q3, with the response flipped due to the opposite sentiment of the question. We examine whether these differences

were significant when controlling for the relatedness of the files and participant-specific effects in Section 5.4.3. For Q4, participants seemed to find *Decision Tree* summaries less helpful, only responding "strongly agree" or "agree" in 28.8% of scenarios. This is surprising, given that *Decision Tree* summaries are widely used in literature, and the *List of Files* baseline is very simple. This potentially suggests that *Decision Tree* summaries present information that distracts users. Future work may wish to examine what aspects of *Decision Tree* summaries are unhelpful and in what situations. Participants indicated that *List of Files* summaries were helpful in 40.5% of scenarios, *Rules-Text* in 56.8% and *Rules-Tree* in 47.3%. The slightly lower rate of positive responses for *Rules-Tree* summaries compared to *Rules-Text* summaries, combined with the similarity in presentation between *Decision Tree* and *Rules-Tree* summaries offers some further evidence that participants considered the decision tree format as a whole less helpful, versus the specific information given in the decision tree nodes. The proportion of positive responses to Q5 for *Rules-Text* and *Rules-Tree* summaries compared to other summary types offers some evidence that such summary types were more verifiable. Participants responded "strongly agree" or "agree" for 68.2% of *Rules-Text* summaries, for 63.7% of *Rules-Tree* summaries, for 49.6% of *List of Files* summaries, and for 45.1% of *Decision Tree* summaries. Interestingly, despite the minimal information in *List of Files* summaries, participants appeared to believe they could still identify which files were covered by the summary.

Additionally, we find that participants indicated stronger confidence in a greater proportion of scenarios for *Rules-Text* or *Rules-Tree* summaries compared to others. Participants responded "strongly agree" or "agree" for 46.2% and 47.9% for *Rules-Text* and *Rules-Tree* summaries, respectively. In contrast, participants only responded such for 25.5% of scenarios with *Decision Tree* summaries and 35.9% of scenarios with *List of Files* summaries. In this case, despite the slightly lower support for *Rules-Tree* summaries indicated in questions such as the helpfulness of the style, *Rules-Tree* summaries were the type that participants found

	Describe (Q2)	Confusing (Q3)	Helpful (Q4)	Verify (Q5)	Related Reason (Q6)	Confident (Q8)
Fixed Effects						
Related (Q1)	1.761***	0.641***	1.673***	2.101***	3.049***	—
Take Recommend (Q7)	——	——	——	——	——	7.592***
Decision Tree	0.553*	2.729***	0.571*	0.945	0.842	0.774
Rules-Text	2.729***	0.353***	2.718***	2.791***	1.032	1.987**
Rules-Tree	1.637*	0.630*	1.412	1.893**	1.013	1.567
Random Effects						
Participant effect	1.169	1.110	1.618	1.318	1.586	1.370

Table 5.4: Cumulative link logit mixed effects regressions on the Likert responses for Summary-Based questions. Coefficients are odds ratios, interpreted as the multiplicative increase in the odds of a higher response. p-values were calculated based on the Satterthwaite method. Asterisks indicate level of statistical significance. (*** = $p < 0.001$, ** = $p < 0.01$, * = $p < 0.05$).

improved their confidence in the most scenarios. The answers to this question go hand-in-hand with the answers to Q5, as both are aimed at determining whether the summaries helped participants make better / more informed decisions with groups of recommendations. We analyze whether this trend held when controlling for other factors below.

5.4.3 Regression Model

To disentangle correlated factors in the responses in Figure 5.3, we built a set of cumulative linked logit mixed effects regression models (Table 5.4). We chose this model format because Likert responses are ordinal and responses by the same participant are correlated.

We take the Likert rating of the "Summary-Based" questions as our response variables. For the models of Q2–Q6, the fixed effects are the presence of each summary type compared against the *List of Files* type, as well as the Likert response to Q1. This last factor is because participants will likely rate summaries more negatively if participants believe the files recommended are less related to each other. For **Confident (Q8)**, the Likert response from Q1 is changed for Q7, indicating whether a participant would accept the group of recommendations in the first place. If a participant is unlikely to accept a group of recommendations, the summary quality is irrelevant to their confidence in accepting the recommendations. We exclude from these models the size of the group of recommendations, and the reason a group

was sampled, as these were not found to be statistically significant factors in any model where they were included. This potentially indicates that our results apply to recommendation groups of a range of sizes and with a variety of properties. Table 5.4 displays odds ratios, which are interpreted as the multiplicative increase in the odds that a higher Likert response is given for the dependent variable when a summary type is present or when the Likert response for a covariate is one point higher. For example, as seen in the first column of Table 5.4, a participant's response was roughly 2.7x more likely to be a higher Likert rating if a *Rules-Text* summary was provided as compared to a *List of Files*.

The only summary type that is statistically significant across all but one model is *Rules-Text*. Further, in each model, the effect direction is as expected: the odds ratio is > 1 (a multiplicative *increase*) for all questions where higher agreement indicates positive attributes, and < 1 for **Confusing (Q3)**, where lower confusion is preferred. The effect size is also notable: the presence of a *Rules-Text* summary has a 2.7× odds improvement for models Q2–Q5, and a 2.0× improvement for **Confident (Q8)**. The effect size, combined with the high statistical significance of the *Rules-Text* summary variable, suggests that such summaries may carry a number of benefits: they may be more understandable, helpful, and confidence-inducing while being less confusing. While *Rules-Tree* summaries also showed some benefit compared to *List of Files* summaries, the effect size and statistical significance were lower. The *Decision Tree* variable in the regression models, when significant, was rated *lower* than baseline *List of Files* summaries: they were less often able to be described (Q2, 0.5×) or to be helpful (Q4, 0.5×) and were more often confusing (Q3, 2.7×). Given that *Rules-Text* and *Rules-Tree* summaries differed only in that *Rules-Tree* presented information like *Decision Tree* summaries did, this suggests that the *Decision Tree* format may require additional improvements to be competitive with other approaches along the same metrics. We leave the specifics of these needed improvements to future work. Interestingly, the sole model where no summary type had a statistically significant effect was **Related**

Reason (Q6). One interpretation is that, though summaries could be effective at helping participants verify inputs, they may have differed from the participants' mental model of the identified files. Future work may wish to examine this effect when summaries are incorporated into full tools. We additionally find that the participant-specific effect for a model was, on average, about a point to a point-and-a-half difference in Likert response. This suggests that even independent of the relatedness of recommendations or the summary type presented, participants still responded to scenarios very differently. This suggests that future work on sets of related recommendations may find significant benefit in personalization of recommendations (McInerney et al. [2018]).

5.5 Summary

In this chapter, we proposed and evaluated a new way of summarizing groups of files being surfaced by an ML-driven recommender system in the context of managing files in cloud storage. We also presented an efficient approximation algorithm to synthesize these summaries. We conducted a 44-participant, within-subjects online user study in which we compared our newly proposed summaries (*Rules-Text* and *Rules-Tree*) against baselines (*List of Files* and *Decision Tree*). Compared to our baselines, participants were more likely to rate *Rules-Text* summaries as more verifiable and as increasing their confidence when managing a groups of recommendations without examining the files individually.

The work of this chapter contributed toward a resolution to one of KondoCloud's drawbacks identified in Chapter 4. Future work will need to investigate affordances that should be used to support summaries in interfaces. Placing such summaries in KondoCloud, for example, would require generating recommendation cards that can include all of the relevant context that was available in our user study.

Future work may also consider generalizing the summarization problem. We implicitly assume that groups of recommendations should be summarized as single units. Potentially,

however, algorithms could be developed to identify an appropriate partition of recommendation groups such that they can be most effectively summarized using multiple distinct summaries. Alternatively, interfaces could empower users to generate this partition and further predicates for themselves. In work like SmallStar (Kurlander et al. [1993]) and Wrangler (Kandel et al. [2011]), users are able to iteratively specify short programs within a given framework. Similar interactions could be made available to users using the structure of *Rules-Text* summaries, offering new modes of interaction in file recommendation settings.

CHAPTER 6

DISCUSSION

The work in this dissertation offers many lesson to future work in these areas. The overall lesson is that of feasibility: the data and techniques are available to offer support for file management actions beyond retrieval. Although we only explored one particular instantiation, in which recommendaions were offered after actions were taken on similar files, similar instantiations are likely possible. Related work, for example, (Khan et al. [2021]) has shown promise in *a prior* identification of files that could have richer actions applied to them. While a direct translation of their techniques is non-trivial (e.g., what features should be collected in order to identify which files are likely to be moved?), future work may take inspiration from the core ideas. For example, similar to Fitchett et al. [2014], interfaces could highlight groups of files likely to be moved to the same location. While the core idea is the same, the affordance is different, and may offer unanticipated benefits.

These ideas could be augmented by enriching the space of file actions available. For example, Bergman et al. [2009] developed a tool that offered users a "deletion-lite" option that separated files deemed to be of lower importance. Similarly, if users are reluctant to take recommendations of difficult-to-reverse actions such as file deletion, offering recommendations of such a deletion-lite action, which can easily be reversed, might improve rates of acceptance. Alternatively, files identified by a system as unlikely to be accessed in the near future could be recommended to be placed in a compressed format. This would save storage space, while still allowing for retrieval after decompression. Many similar ideas are available, and could easily adopt the same format of file recommendations as used in KondoCloud.

In order to fully translate the techniques here to practical systems, many improvements can be made. As identified in Section 4.6, the limitations of KondoCloud in improving scalability, modeling task context, and enabling richer interaction remain open. Additionally, though the work of Chapter 5 followed on the identified drawback of KondoCloud from

Section 4.6, we did not incorporate it fully into the KondoCloud interface. Exploring how summarization can be incorporated into an interface will be another interesting direction for future work. Fully realizing these will be important in improving the state of art for file management in personal cloud storage.

CHAPTER 7

ADDITIONAL ARTIFACTS

7.1 Additional Artifacts A: Investigation Study Survey Instrument

7.1.1 Part 1

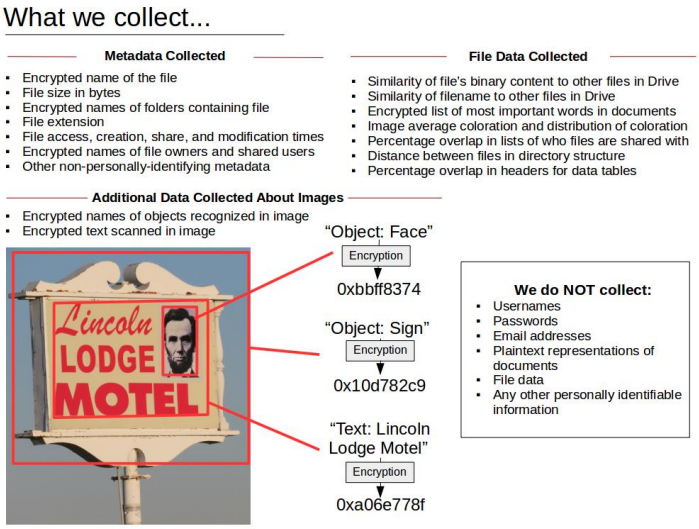
As part of this study, our computer code collects some data from your Google Drive or Dropbox account. As shown in the image below, it collects some file metadata as well as encrypted text from documents or images and objects recognized in images. It does not save any of the files. Researchers will never have access to the human-readable versions of any file content.

To access these files, we ask you to log in through the secure OAuth service. This provides our code with a one-time access token to use for this study. We do not save any usernames, passwords, or personally identifiable information through this process. If you are interested, you can read more about how Google allows third parties to access your account, and how you can manage this access here: `https://support.google.com/accounts/bin/answer.py?hl=en&answer=143031`. You may find equivalent information for Dropbox here: `https://www.dropbox.com/help/security/third-party-apps`.

If these terms are acceptable, please indicate so below. Otherwise, you will be asked to release the submission.

1. I agree to provide access to my Google Drive or Dropbox account under the terms specified above.

    - Yes
    - No



Demographics

1. Are you the only person with access to this account, or do you share the username and password to this account with others?

   - I am the only person with access
   - Other users have acces
   - Not sure

2. To the best of your knowledge, how many users (including yourself) have access to this account?

3. For what purpose(s) do you use this account?

4. What percentage of the data in this account would you characterize to be primarily for personal use?

5. What percentage of the data in this account would you characterize to be primarily for professional use? (i.e., related to your job or career)

6. How do you interact with your *[Cloud Service]* account? Please mark all that apply.

- I use the website
- I use the app
- I sync it with folders on my computer

7. How often do you open the website or app for your *[Cloud Service]* account?

   - Daily
   - Weekly
   - Monthly
   - Yearly or less
   - I don't know

8. Please rate your agreement with the following statement: "My *[Cloud Service]* account is well-organized"

   - Strongly agree
   - Agree
   - Neutral
   - Disagree
   - Strongly disagree
   - I don't know

9. Please describe your strategies for organizing your *[Cloud Service]* account in 5 or fewer sentences.

10. Do you use folders to organize your account?

    - Yes

- No
- I don't know

11. (If 'Yes' to previous question) How?

12. (If 'No' to previous question) Why not?

13. Please describe a specific experience in which you were <u>not</u> able to find a file you were looking for in your *[Cloud Service]* account. If you have not had such an experience, please state so.

14. Please list any other cloud storage services (e.g. Sharepoint, Box, iCloud) that you use personally or professionally.

15. With what gender do you identify?

    - Male
    - Female
    - Non-binary / other
    - I prefer not to answer

16. Are you majoring in, hold a degree in, or have held a job in any of the following fields: computer science; computer engineering; information technology; or a related field?

    - Yes
    - No
    - I don't know

17. What is your age range?

    - 18-24 years old

- 25-34 years old
- 35-44 years old
- 45-54 years old
- 55-64 years old
- 65 years or older
- Prefer not to answer

18. What is your occupation? (optional)

7.1.2 Part 2

Instructions

Our research goal is to design systems to help people manage their cloud storage accounts. We will present to you 18 pairs of files from your *[Cloud Service]* account, and ask you to rate their similarity based on the following categories. You will be able to return to these instructions at any point during the survey.

The Data Itself

- **Topic**–two files are similar if they talk about the same subject matter
  - Example: a photo of a dog and a document about dog grooming techniques

Origin

- **Purpose**–two files are similar if they will likely be used for similar tasks or goals
  - Example: a photo of a dog and a document about dog grooming techniques
- **Derivation**–two files are similar if they are different versions of the same item, or if one "created" the other

  - Example: a rough draft of a proposal document, and a final version of the same document
  - Example: a music score, and a recording of you playing the music from that score

- **Creation Context**–two files are similar if they were created at the same time or in the same place
  - Example: a short story you wrote at a writer's retreat, and another person's poem written at the same retreat

Content

(This section is repeated 18 times, each with a different file pair)

You are on pair *[Current File Pair]* out of 18 file pairs.

Please answer the following questions in reference to the following two files:

File 1: *[File Name A] ([Preview Link])*

File 2: *[File Name B] ([Preview Link])*

Note that you must preview the files at the provided links to continue in the survey.

To review the tutorial: *[Link to Tutorial]*

1. Please give a short description of **File 1**

2. Please give a short description of **File 2**

3. Please describe in general how these files are similar or dissimilar.

4. I consider these two files to be similar in Topic.

    - Strongly agree
    - Agree
    - Neutral
    - Disagree
    - Strongly disagree
    - I don't know

5. I consider these two files to be similar in Derivation.

    - Strongly agree

- Agree
- Neutral
- Disagree
- Strongly disagree
- I don't know

6. I consider these two files to be similar in Purpose.

   - Strongly agree
   - Agree
   - Neutral
   - Disagree
   - Strongly disagree
   - I don't know

7. I consider these two files to be similar in Creation Context.

   - Strongly agree
   - Agree
   - Neutral
   - Disagree
   - Strongly disagree
   - I don't know

8. It is okay if all copies of **File 1** are deleted.

   - Strongly agree

- Agree
- Neutral
- Disagree
- Strongly disagree
- I don't know

9. It is okay if all copies of **File 2** are deleted.

   - Strongly agree
   - Agree
   - Neutral
   - Disagree
   - Strongly disagree
   - I don't know

10. I would be upset if the contents of **File 1** were to be released publicly.

    - Strongly agree
    - Agree
    - Neutral
    - Disagree
    - Strongly disagree
    - I don't know

11. I would be upset if the contents of **File 2** were to be released publicly.

    - Strongly agree

- Agree
- Neutral
- Disagree
- Strongly disagree
- I don't know

(Page break)

Please answer the following questions in reference to the following two files:

File 1: *[File Name A] ([Preview Link])*

File 2: *[File Name B] ([Preview Link])*

To review the tutorial: *[Link to Tutorial]*

12. If I were searching for information, and I found one of these files to be relevant, I would also want to see the other file.
    - Strongly agree
    - Agree
    - Neutral
    - Disagree
    - Strongly disagree
    - I don't know
13. If I were organizing my *[Cloud Service]* account, and I wanted to move one of these files to a new location, I would also want to move the other file to that same location.

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree
- I don't know

14. If I were organizing my *[Cloud Service]* account, and I wanted to delete one of these files, I would also want to delete the other file.

    - Strongly agree
    - Agree
    - Neutral
    - Disagree
    - Strongly disagree
    - I don't know

15. Which of the below was/were MOST informative in your answers to the previous 3 questions? (Please mark all that apply.)

    - The similarity of the files' Topic
    - The similarity of the files' Derivation
    - The similarity of the files' Purpose
    - The similarity of the files' Creation Context
    - I don't know

7.2 Additional Artifacts B: Chapter 3 Full Mixed Effects Ordinal Regression Models

7.2.1 Topic

	Estimate	Std Err.	Odds Ratio	z-value	p-value	
Last Modified	3.033	0.382	20.769	7.947	0.000	***
Filename	1.381	0.470	3.978	2.936	0.003	**
File Size	−0.015	0.351	0.985	−0.042	0.967	
Tree Distance	−0.754	0.746	0.471	−1.010	0.313	
Shared Users	0.887	0.307	2.428	2.891	0.004	**
File Contents	1.049	0.326	2.855	3.221	0.001	**
Text Topic	1.122	0.327	3.072	3.432	0.001	***
Table Schema	1.377	1.144	3.965	1.204	0.229	
Image Contents	3.605	0.683	36.777	5.280	0.000	***
Deep Hierarchy	−0.123	0.392	0.884	−0.315	0.753	

7.2.2 Purpose

	Estimate	Std Err.	Odds Ratio	z-value	p-value	
Last Modified	2.436	0.364	11.422	6.686	0.000	***
Filename	2.542	0.452	12.699	5.625	0.000	***
File Size	0.541	0.317	1.718	1.706	0.088	
Tree Distance	−0.531	0.689	0.588	−0.771	0.441	
Shared Users	1.056	0.294	2.874	3.595	0.000	***
File Contents	1.245	0.320	3.473	3.894	0.000	***
Text Topic	0.952	0.307	2.592	3.103	0.002	**
Table Schema	2.571	1.094	13.076	2.350	0.019	*
Image Contents	3.368	0.659	29.018	5.110	0.000	***
Deep Hierarchy	0.227	0.277	1.255	0.819	0.413	

7.2.3 Derivation

	Estimate	**Std Err.**	**Odds Ratio**	**z-value**	**p-value**	
Last Modified	1.165	0.410	3.207	2.842	0.004	**
Filename	2.896	0.501	18.094	5.775	0.000	***
File Size	0.604	0.401	1.829	1.506	0.132	
Tree Distance	−0.147	0.849	0.863	−0.173	0.863	
Shared Users	1.050	0.322	2.857	3.262	0.001	**
File Contents	1.428	0.339	4.172	4.212	0.000	***
Text Topic	0.840	0.329	2.315	2.553	0.011	*
Table Schema	0.612	1.410	1.845	0.434	0.664	
Image Contents	2.190	0.654	8.938	3.349	0.001	***
Deep Hierarchy	0.075	0.527	1.078	0.142	0.887	

7.2.4 Creation

	Estimate	**Std Err.**	**Odds Ratio**	**z-value**	**p-value**	
Last Modified	2.838	0.396	17.075	7.173	0.000	***
Filename	2.550	0.477	12.805	5.342	0.000	***
File Size	0.410	0.373	1.507	1.097	0.272	
Tree Distance	0.403	0.754	1.496	0.534	0.593	
Shared Users	1.139	0.333	3.124	3.424	0.001	***
File Contents	1.309	0.343	3.703	3.817	0.000	***
Text Topic	0.815	0.318	2.260	2.563	0.010	*
Table Schema	0.873	1.279	2.393	0.682	0.495	
Image Contents	2.170	0.657	8.757	3.300	0.001	***
Deep Hierarchy	0.711	0.416	2.036	1.709	0.087	

7.2.5 Find

	Estimate	Std Err.	Odds Ratio	z-value	p-value	
Last Modified	2.488	0.369	12.034	6.736	0.000	***
Filename	2.387	0.461	10.885	5.175	0.000	***
File Size	0.321	0.366	1.379	0.878	0.380	
Tree Distance	−0.893	0.746	0.409	−1.196	0.232	
Shared Users	1.922	0.319	6.833	6.033	0.000	***
File Contents	0.931	0.314	2.536	2.961	0.003	**
Text Topic	0.535	0.316	1.707	1.693	0.091	
Table Schema	1.066	1.189	2.905	0.897	0.370	
Image Contents	2.571	0.639	13.085	4.026	0.000	***
Deep Hierarchy	0.326	0.483	1.385	0.675	0.500	

7.2.6 Move

	Estimate	Std Err.	Odds Ratio	z-value	p-value	
Last Modified	2.535	0.367	12.618	6.908	0.000	***
Filename	2.621	0.467	13.744	5.609	0.000	***
File Size	0.505	0.353	1.656	1.430	0.153	
Tree Distance	−0.693	0.768	0.500	−0.903	0.367	
Shared Users	1.977	0.322	7.218	6.137	0.000	***
File Contents	0.707	0.304	2.027	2.322	0.020	*
Text Topic	0.951	0.330	2.588	2.886	0.004	**
Table Schema	0.668	1.168	1.951	0.572	0.567	
Image Contents	2.313	0.664	10.106	3.483	0.000	***
Deep Hierarchy	−0.351	0.435	0.704	−0.806	0.420	

7.2.7 Delete

	Estimate	**Std Err.**	**Odds Ratio**	**z-value**	**p-value**	
Last Modified	2.031	0.359	7.623	5.660	0.000	***
Filename	1.455	0.441	4.286	3.302	0.001	***
File Size	0.802	0.340	2.231	2.361	0.018	*
Tree Distance	0.086	0.735	1.090	0.117	0.907	
Shared Users	1.723	0.308	5.604	5.599	0.000	***
File Contents	0.787	0.301	2.197	2.618	0.009	**
Text Topic	0.423	0.305	1.526	1.384	0.166	
Table Schema	1.385	1.100	3.993	1.258	0.208	
Image Contents	1.018	0.596	2.767	1.706	0.088	
Deep Hierarchy	−0.237	0.483	0.789	−0.491	0.623	

REFERENCES

David Abrams, Ron Baecker, and Mark Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1998.

Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

Nitin Agrawal, William J. Bolosky, John R. Douceur, and Jacob R. Lorch. A five-year study of file-system metadata. *ACM TOS*, 3(3):9, 2007.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

Nehad Albadri, Richard Watson, and Stijn Dekeyser. Treetags: bringing tags to the hierarchical file system. In *Proceedings of the Australasian Computer Science Week Multiconference*, pages 1–10, 2016.

Tarfah Alrashed, Ahmed Hassan Awadallah, and Susan Dumais. The lifetime of email messages: A large-scale analysis of email revisitation. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, 2018.

Tarfah Alrashed, Chia-Jung Lee, Peter Bailey, Christopher Lin, Milad Shokouhi, and Susan Dumais. Evaluating user actions as a proxy for email significance. In *Proceedings of the World Wide Web Conference*, 2019.

Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, et al. Guidelines for human-AI interaction. In *Proc. CHI*, 2019.

Anne Aula, Natalie Jhaveri, and Mika Käki. Information search and re-access strategies of experienced web users. In *Proc. WWW*, 2005.

Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, 2009.

Xinlong Bao and Thomas G. Dietterich. Folderpredictor: Reducing the cost of reaching the right folder. *ACM TIST*, 2(1), 2011.

Xinlong Bao, Jonathan L. Herlocker, and Thomas G. Dietterich. Fewer clicks and less frustration: Reducing the cost of reaching the right folder. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, 2006.

Deborah Barreau. Context as a factor in personal information management systems. *Journal of the American Society for Information Science*, 46(5):327–339, 1995a.

Deborah Barreau and Bonnie A. Nardi. Finding and reminding: File organization from the desktop. *ACM SIGCHI Bulletin*, 27(3):39–43, 1995.

Deborah K. Barreau. Context as a factor in personal information management systems. *Journal of the American Society for Information Science*, 46(5), 1995b.

Nicholas J Belkin. Anomalous states of knowledge as a basis for information retrieval. *Canadian journal of information science*, 5(1):133–143, 1980.

Yael Benn, Ofer Bergman, Liv Glazer, Paris Arent, Iain D. Wilkinson, Rosemary Varley, and Steve Whittaker. Navigating through digital folders uses the same brain structures as real world navigation. *Scientific Reports*, 5(1), 2015.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of the KDD Cup and Workshop*, 2007.

Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. The user-subjective approach to personal information management systems. *Journal of the American Society for Information Science and Technology*, 54(9):872–878, 2003.

Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. The user-subjective approach to personal information management systems design: Evidence and implementations. *JASIST*, 59(2):235–246, 2008a.

Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. Improved search engines and navigation preference in personal information management. *ACM TOIS*, 26(4), 2008b.

Ofer Bergman, Simon Tucker, Ruth Beyth-Marom, Edward Cutrell, and Steve Whittaker. It's not that important: demoting personal information of low subjective importance using grayarea. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009.

Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. The effect of folder structure on personal file navigation. *Journal of the American Society for Information Science and Technology*, 61(12):2426–2441, 2010.

Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. How do we find personal files? the effect of os, presentation & depth on file navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.

Ofer Bergman, Noa Gradovitch, Judit Bar-Ilan, and Ruth Beyth-Marom. Folder versus tag preference in personal information management. *Journal of the American Society for Information Science and Technology*, 64(10):1995–2012, 2013a.

Ofer Bergman, Noa Gradovitch, Judit Bar-Ilan, and Ruth Beyth-Marom. Tagging personal information: A contrast between attitudes and behavior. In *Proceedings of the 76th ASIS&T Annual Meeting*, 2013b.

Ofer Bergman, Maskit Tene-Rubinstein, and Jonathan Shalom. The use of attention resources in navigation versus search. *Personal and Ubiquitous Computing*, 17(3):583–590, 2013c.

Ofer Bergman, Steve Whittaker, and Yaron Frishman. Let's get personal: The little nudge that improves document retrieval in the cloud. *J. Doc*, 2019.

Mustafa Bilgic and Raymond J Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond personalization workshop, IUI*, volume 5, page 153, 2005.

Richard Boardman and M. Angela Sasse. Stuff goes into the computer and doesn't come out: A cross-tool study of personal information management. In *Proc. CHI*, 2004.

Richard Boardman, Robert Spence, and M. Angela Sasse. Too many hierarchies? the daily struggle for control of the workspace. In *Proc. HCII*, 2003.

Christian Borgelt. Frequent item set mining. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 2(6):437–456, 2012.

Sergio Canuto, Thiago Salles, Thierson C. Rosa, and Marcos A. Gonçalves. Similarity-based synthetic document representations for meta-feature generation in text classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

Robert Capra and M.A. Perez-Quinones. Factors and evaluation of refinding behaviors. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 175–182, 2016.

Surajit Chaudhuri and Luis Gravano. Evaluating top-k selection queries. In *Vldb*, volume 99, pages 397–410. Citeseer, 1999.

Li Chen and Pearl Pu. Preference-based organization interfaces: aiding user critiques in recommender systems. In *International Conference on User Modeling*, pages 77–86. Springer, 2007.

Li Chen and Feng Wang. Explaining recommendations based on feature sentiments in product reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 17–28, 2017.

Suming Jeremiah Chen, Zhen Qin, Zachary Teal Wilson, Brian Lee Calaci, Michael Richard Rose, Ryan Lee Evans, Sean Robert Abraham, Don Metzler, Sandeep Tata, and Mike Colagrosso. Improving recommendation quality at google drive. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

Yen-Liang Chen and Lucas Tzu-Hsuan Hung. Using decision trees to summarize associative classification rules. *Expert Systems with Applications*, 36(2):2338–2351, 2009.

Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Beagle++: Semantically enhanced searching and ranking on the desktop. In *Proceedings of the European Semantic Web Conference*, 2006.

Andrea Civan, William Jones, Predrag Klasnja, and Harry Bruce. Better to organize personal information by folders or by tags?: The devil is in the details. *Proceedings of the American Society for Information Science and Technology*, 45(1), 2008.

Jason W. Clark, Peter Snyder, Damon McCoy, and Chris Kanich. I saw images i didn't even know i had: Understanding user perceptions of cloud storage privacy. In *Proc. CHI*, 2015.

Andy Cockburn and Bruce McKenzie. What do web users do? an empirical analysis of web use. *IJHCS*, 54(6):903–922, 2001.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18(5):455–496, 2008.

Gabor Cselle, Keno Albrecht, and Rogert Wattenhofer. Buzztrack: topic detection and tracking in email. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 190–197, 2007.

Marek Czarkowski and Judy Kay. A scrutable adaptive hypertext. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 384–387. Springer, 2002.

Mary Czerwinski, Eric Horvitz, and Susan Wilhite. A diary study of task switching and interruptions. In *Proc. CHI*, 2004.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*, 2020.

Sanjoy Dasgupta and Sivan Sabato. Robust learning from discriminative feature feedback. In *Proc. AISTATS*, 2020.

Jesse David Dinneen and Ilja Frissen. Mac users do it differently: the role of operating system and individual differences in file management. In *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.

Jesse David Dinneen and Charles-Antoine Julien. The ubiquitous digital file: A review of file management research. *Journal of the Association for Information Science and Technology*, 2019.

Jesse David Dinneen, Fabian Odoni, Ilja Frissen, and Charles-Antoine Julien. Cardinal: Novel software for studying file management behavior. In *Proceedings of the 79th ASIS&T Annual Meeting*, 2016.

Jesse David Dinneen, Charles-Antoine Julien, and Ilja Frissen. The scale and structure of personal file collections. In *Proc. CHI*, 2019.

Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. Caep: Classification by aggregating emerging patterns. In *International Conference on Discovery Science*, pages 30–42. Springer, 1999.

Paul Dourish. The appropriation of interactive technologies: Some lessons from placeless documents. *CSCW*, 12(4):465–490, 2003.

Paul Dourish, W. Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen, Michael Salisbury, Douglas B. Terry, and James Thornton. Extending document management systems with user-specific active properties. *ACM TOIS*, 18(2):140–170, 2000.

Dropbox. `https://help.dropbox.com/files-folders/sort-preview/multi-file-organize`, 2022.

Susan Dumais, Edward Cutrell, Jonathan J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *Proc. SIGIR*, 2003.

Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.

Kareem El Gebaly, Parag Agrawal, Lukasz Golab, Flip Korn, and Divesh Srivastava. Interpretable and informative explanations of outcomes. *Proceedings of the VLDB Endowment*, 8(1):61–72, 2014.

Alexander Felfernig and Bartosz Gula. An empirical study on consumer behavior in the interaction with knowledge-based recommender applications. In *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)*, pages 37–37. ieee, 2006.

Leah Findlater and Joanna McGrenere. Beyond performance: Feature awareness in personalized interfaces. *IJHCS*, 68(3):121–137, 2010.

Stephen Fitchett and Andy Cockburn. Accessrank: Predicting what users will do next. In *Proc. CHI*, 2012.

Stephen Fitchett, Andy Cockburn, and Carl Gutwin. Finder highlights: Field evaluation and design of an augmented file browser. In *Proc. CHI*, 2014.

Eric Freeman and David Gelernter. Lifestreams: A storage model for personal data. *ACM SIGMOD Record*, 25(1):80–86, 1996.

Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI*, 2006.

Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. Predictability and accuracy in adaptive user interfaces. In *Proc. CHI Extended Abstracts*, 2008.

Qin Gao. An empirical study of tagging for personal information organization: Performance, workload, memory, and consistency. *International Journal of Human-Computer Interaction*, 27(9):821–863, 2011.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proc. CVPR*, 2013.

Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.

Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proc. VLDB*, 1999.

Google. Vision ai. `https://cloud.google.com/vision/`, 2019.

Julien Gori, Han L. Han, and Michel Beaudouin-Lafon. Fileweaver: Flexible file management with automatic dependency tracking. In *Proc. UIST*, 2020.

Robert Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

Saul Greenberg and Ian H. Witten. Adaptive personalized interfaces—a question of viability. *Behaviour & Information Technology*, 4(1):31–45, 1985.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018a.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018b.

Karl Gyllstrom. *Enriching personal information management with document interaction histories*. PhD thesis, 2009.

Sharon Hardof-Jaffe, Arnon Hershkovitz, Hama Abu-Kishk, Ofer Bergman, and Rafi Nachmias. Students' organization strategies of personal information space. *Journal of Digital Information*, 10(5), 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

Jeffrey Heer. Agency plus automation: Designing artificial intelligence into interactive systems. *PNAS*, 116(6):1844–1850, 2019.

Sarah Henderson and Ananth Srinivasan. An empirical analysis of personal digital document structures. In *Proc. HCI*, 2009.

Jonathan Lee Herlocker. *Understanding and improving automated collaborative filtering systems*. University of Minnesota, 2000.

MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

Farnaz Jahanbakhsh, Ahmed Hassan Awadallah, Susan T. Dumais, and Xuhai Xu. Effects of past interactions on user experience with recommended documents. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, 2020.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE TPAMI*, 33(1):117–128, 2010.

Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. Interactive data exploration with smart drill-down. *IEEE Transactions on Knowledge and Data Engineering*, 31(1):46–60, 2017.

William Jones, Susan Dumais, and Harry Bruce. Once found, what then? a study of "keeping" behaviors in the personal use of web information. *Proceedings of the American Society for Information Science and Technology*, 39(1):391–402, 2002.

William Jones, Ammy Jiranida Phuwanartnurak, Rajdeep Gill, and Harry Bruce. Don't take my folders away! organizing personal information to get things done. In *Proc. CHI Extended Abstracts*, 2005.

Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 3363–3372, 2011.

Taylor Kessler Faulkner, Will Brackenbury, and Ashwin Lall. K-regret queries with nonlinear utilities. *Proceedings of the VLDB Endowment*, 8(13):2098–2109, 2015.

Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. Forgotten but not gone: Identifying the need for longitudinal data management in cloud storage. In *Proc. CHI*, 2018.

Mohammad Taha Khan, Christopher Tran, Shubham Singh, Dimitri Vasilkov, Chris Kanich, Blase Ur, and Elena Zheleva. Helping users automatically find and manage sensitive, expendable files in cloud storage. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1145–1162, 2021.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

Carol C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *JASIST*, 42(5):361–371, 1991.

David Kurlander, Allen Cypher, and Daniel Conrad Halbert. *Watch what I do: programming by demonstration*. MIT press, 1993.

Barbara H. Kwasnik. How a personal document's intended use or purpose affects its classification in an office. In *ACM SIGIR Forum*, volume 23, pages 207–210, 1989.

Barbara H. Kwasnik. The importance of factors that are not document attributes in the organization of personal documents. *J. Doc*, 1991.

Barbara H. Kwasnik. The role of classification structures in reflecting and building theory. *Advances in Classification Research Online*, 3(1):63–82, 1992.

Mark W. Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, 1988.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

Paul J. Lavrakas. *Encyclopedia of survey research methods*. Sage Publications, 2008.

Bongshin Lee and Benjamin B. Bederson. Favorite folders: A configurable, scalable file browser. Technical report, 2003.

John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80, 2004.

Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE international conference on data mining*, pages 369–376. IEEE, 2001.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proc. IUI*, 2010.

Wanyu Liu, Olivier Rioul, Joanna Mcgrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. Bigfile: Bayesian information gain for fast file retrieval. In *Proc. CHI*, 2018.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

Joel Mackenzie, Kshitiz Gupta, Fang Qiao, Ahmed Hassan Awadallah, and Milad Shokouhi. Exploring user behavior in email re-finding tasks. In *The World Wide Web Conference*, pages 1245–1255, 2019.

Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM TOIS*, 1(1):99–112, 1983.

Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. Tell me what i need to know: succinctly summarizing data with itemsets. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 573–581, 2011.

Gary Marsden and David E. Cairns. Improving the usability of the hierarchical file system. *South African Computer Journal*, 2004(32):69–78, 2004.

Tony Mason and Margo Seltzer. Not dead yet: Hierarchical file systems won't die. 2019.

Charlotte Massey, Thomas Lennig, and Steve Whittaker. Cloudy forecast: an exploration of the factors underlying shared repository use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2461–2470, 2014a.

Charlotte Massey, Sean TenBrook, Chaconne Tatum, and Steve Whittaker. Pim and personality: What do our personal file systems say about us? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014b.

Ben McCamish, Vahid Ghadakchi, Arash Termehchy, Behrouz Touri, and Liang Huang. The data interaction game. In *Proc. SIGMOD*, 2018.

Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. Thinking positively-explanatory feedback for conversational recommender systems. In *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*, pages 115–124, 2004.

James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM conference on recommender systems*, pages 31–39, 2018.

Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *The World Wide Web Conference*, pages 1256–1267, 2019.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, 2013.

David R. Millen, Jonathan Feinberg, and Bernard Kerr. Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006.

David R. Millen, Meng Yang, Steven Whittaker, and Jonathan Feinberg. Social bookmarking and exploratory search. In *Proceedings of the 10th European Conference on Computer-Supported Cooperative Work*. 2007.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

Danupon Nanongkai, Ashwin Lall, Atish Das Sarma, and Kazuhisa Makino. Interactive regret minimization. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 109–120, 2012.

Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018.

Xi Niu and Ahmad Al-Doulat. Luckyfind: Leveraging surprise to improve user satisfaction and inspire curiosity in a recommender system. In *Proc. CHIIR*, 2021.

Kyong Eun Oh. What happens once you categorize files into folders? *Proceedings of the American Society for Information Science and Technology*, 49(1):1–4, 2012.

Kyong Eun Oh. Types of personal information categorization: Rigid, fuzzy, and flexible. *JASIST*, 68(6):1491–1504, 2017.

Kyong Eun Oh and Nicholas J. Belkin. Understanding what personal information items make categorization difficult. *Proceedings of the American Society for Information Science and Technology*, 51(1):1–3, 2014.

Michael Oppermann, Robert Kincaid, and Tamara Munzner. Vizcommender: Computing text-based similarity in visualization repositories for content-based recommendations. *arXiv:2008.07702*, 2020.

Soya Park, Amy X. Zhang, Luke S. Murray, and David R. Karger. Opportunities for automating email processing: A need-finding study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2019.

Eyal Peer, Laura Brandimarte, Sonam Samat, and Alessandro Acquisti. Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163, 2017.

Peter Pirolli. *Information foraging theory: Adaptive interaction with information*. Oxford University Press, 2007.

Prolific. `https://www.prolific.co/`, 2019.

Dennis Quan, David Huynh, and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In *Proceedings of the International Semantic Web Conference*, 2003.

Bradley Rhodes and Thad Starner. Remembrance agent: A continuously running automated information retrieval system.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

T.J. Robertson, Shrinu Prabhakararao, Margaret Burnett, Curtis Cook, Joseph R. Ruthruff, Laura Beckwith, and Amit Phalgune. Impact of interruption style on end-user debugging. In *Proc. CHI*, 2004.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

Ali Sajedi, Seyyed Hamidreza Afzali, and Zahra Zabardast. Can you retrieve a file on the computer in your first attempt? think to a new file manager for multiple categorization of your personal information. In *6th international workshop on personal information management*. ACM, 2012.

Leo Sauermann, Gunnar Aastrand Grimnes, Malte Kiesel, Christiaan Fluit, Heiko Maus, Dominik Heim, Danish Nadeem, Benjamin Horak, and Andreas Dengel. Semantic desktop

2.0: The gnowsis experience. In *Proceedings of the International Semantic Web Conference*, 2006.

Markus Schröder, Christian Jilek, and Andreas Dengel. Interactive concept mining on personal data. *arXiv:1903.05872*, 2019.

Andrew Sears and Ben Shneiderman. Split menus: effectively using selection frequency to organize menus. *ACM TOCHI*, 1(1):27–51, 1994.

Richard B. Segal and Jeffrey O. Kephart. Mailcat: An intelligent assistant for organizing e-mail. In *Proc. AGENTS*, 1999.

Procheta Sen, Debasis Ganguly, and Gareth JF Jones. I know what you need: Investigating document retrieval effectiveness with partial session contexts. *ACM Transactions on Information Systems (TOIS)*, 40(3):1–30, 2021.

Amit Sharma and Dan Cosley. Do social explanations work? studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144, 2013.

Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. Programs as black-box explanations. *arXiv preprint arXiv:1611.07579*, 2016.

Debmalya Sinha and Anupam Basu. Gardener: A file browser assistant to help users maintaining semantic folder hierarchy. In *Proc. IHCI*, 2012.

Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831, 2002.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proc. COLT*, 2011.

Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3), 2017.

Studio 42. `https://studio-42.github.io/elFinder/`, 2019.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems*, 30, 2017.

Kumiko Tanaka-Ishii and Ian Frank. Multi-agent explanation strategies in real-time domains. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 158–165, 2000.

John C Tang, Eric Wilcox, Julian A Cerruti, Hernan Badenes, Stefan Nusser, and Jerald Schoudt. Tag-it, snag-it, or bag-it: combining tags, threads, and folders in e-mail. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 2179–2194. 2008.

Sandeep Tata, Alexandrin Popescul, Marc Najork, Mike Colagrosso, Julian Gibbons, Alan Green, Alexandre Mah, Michael Smith, Divanshu Garg, Cayden Meyer, and Reuben Kan. Quick access: building a smart experience for Google Drive. In *Proc. KDD*, 2017.

Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proc. CHI*, 2004.

Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4):399–439, 2012.

Nava Tintarev and Judith Masthoff. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, pages 353–382. Springer, 2015.

Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

Tuan A Tran, Sven Schwarz, Claudia Niederée, Heiko Maus, and Nattiya Kanhabua. The forgotten needle in my collections: Task-aware ranking of documents in semantic information space. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, pages 13–22, 2016.

Usability.gov. `https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html`, 2021.

Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56, 2009.

Francesco Vitale, Izabelle Janzen, and Joanna McGrenere. Hoarding and minimalism: Tendencies in digital data preservation. In *Proc. CHI*, 2018.

Amy Voida, Judith S. Olson, and Gary M. Olson. Turbulence in the clouds: Challenges of cloud-based information work. In *Proc. CHI*, 2013.

Stephen Voida and Elizabeth D. Mynatt. It feels better than filing: Everyday work experiences in an activity-based computing system. In *Proc. CHI*, 2009.

Chao Wang and Srinivasan Parthasarathy. Summarizing itemset patterns using probabilistic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 730–735, 2006.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial intelligence and statistics*, pages 1013–1022. PMLR, 2015.

Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to hash for indexing big data—a survey. *Proceedings of the IEEE*, 104(1):34–57, 2015.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. Interactive summarization and exploration of top aggregate query answers. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 2196. NIH Public Access, 2018.

Roger Whitham and Leon Cruickshank. The function and future of the folder. *Interacting with Computers*, 29(5):629–647, 2017.

Steve Whittaker. Personal information management: From information consumption to curation. *Annual Review of Information Science and Technology*, 45(1):1–62, 2011.

Steve Whittaker and Charlotte Massey. Mood and personal information management: How we feel influences how we organize our information. *Personal and Ubiquitous Computing*, 24(1):695–707, 2020.

Steve Whittaker, Ofer Bergman, and Paul Clough. Easy on that trigger dad: A study of long term family photo retrieval. *Personal and Ubiquitous Computing*, 14(1):31–43, 2010.

Brian Whitworth. Polite computing. *Behaviour & Information Technology*, 24(5):353–363, 2005.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26(3), 2008.

Xuhai Xu, Ahmed Hassan Awadallah, Susan T. Dumais, Farheen Omar, Bogdan Popp, Robert Rounthwaite, and Farnaz Jahanbakhsh. Understanding user behavior for document recommendation. In *Proceedings of The Web Conference*, 2020.

Xifeng Yan, Hong Cheng, Jiawei Han, and Dong Xin. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 314–323, 2005.

Liang Huai Yang, Jian Zhou, Jiacheng Wang, and Mong-Li Lee. A novel PIM system and its effective storage compression scheme. *JSW*, 7(6):1385–1392, 2012.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336, 2017.

Hong Zhang and Xiao Hu. A quantitative comparison on file folder structures of two groups of information workers. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 485–486. IEEE Press, 2014.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92, 2014.

Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, 2018.