

KondoCloud: Improving Information Management in Cloud Storage via Recommendations Based on File Similarity

Will Brackenbury
wbrackenbury@uchicago.edu
University of Chicago

Andrew McNutt
mcnutt@uchicago.edu
University of Chicago

Kyle Chard
chard@uchicago.edu
University of Chicago

Aaron J. Elmore
aelmore@uchicago.edu
University of Chicago

Blase Ur
blase@uchicago.edu
University of Chicago

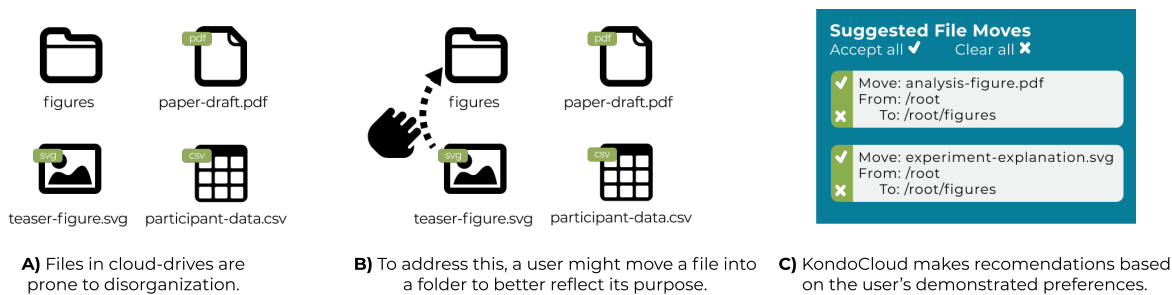


Figure 1: KondoCloud is a file-browsing interface that helps users organize cloud repositories (e.g., Google Drive) by providing ML-based recommendations for files they may want to move, delete, or retrieve based on past actions on similar files.

ABSTRACT

Users face many challenges in keeping their personal file collections organized. While current file-management interfaces help users retrieve files in disorganized repositories, they do not aid in organization. Pertinent files can be difficult to find, and files that should have been deleted may remain. To help, we designed KondoCloud, a file-browser interface for personal cloud storage. KondoCloud makes machine learning-based recommendations of files users may want to retrieve, move, or delete. These recommendations leverage the intuition that similar files should be managed similarly.

We developed and evaluated KondoCloud through two complementary online user studies. In our Observation Study, we logged the actions of 69 participants who spent 30 minutes manually organizing their own Google Drive repositories. We identified high-level organizational strategies, including moving related files to newly created sub-folders and extensively deleting files. To train the classifiers that underpin KondoCloud's recommendations, we had participants label whether pairs of files were similar and whether they should be managed similarly. In addition, we extracted ten metadata and content features from all files in participants' repositories. Our logistic regression classifiers all achieved F1 scores of 0.72 or higher. In our Evaluation Study, 62 participants used KondoCloud either

with or without recommendations. Roughly half of participants accepted a non-trivial fraction of recommendations, and some participants accepted nearly all of them. Participants who were shown the recommendations were more likely to delete related files located in different directories. They also generally felt the recommendations improved efficiency. Participants who were not shown recommendations nonetheless manually performed about a third of the actions that would have been recommended.

CCS CONCEPTS

• Information systems → Data management systems.

KEYWORDS

cloud storage, recommendations, Google Drive, personal information management

ACM Reference Format:

Will Brackenbury, Andrew McNutt, Kyle Chard, Aaron J. Elmore, and Blase Ur. 2021. KondoCloud: Improving Information Management in Cloud Storage via Recommendations Based on File Similarity. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*, October 10–14, 2021, Virtual Event, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3472749.3474736>

1 INTRODUCTION

Users store large amounts of data in personal cloud storage systems like Google Drive and Dropbox [35]. Unfortunately, users' cloud repositories often end up disorganized. Because information can be difficult to categorize [76], fragmented across services [11, 20], or filed by others who organize differently [16], files are often uncategorized [64] or stored in unsuitable locations [74]. Re-finding

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
UIST '21, October 10–14, 2021, Virtual Event, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8635-7/21/10.
<https://doi.org/10.1145/3472749.3474736>

files becomes difficult [18, 92]. Unneeded, and often sensitive, files accumulate [25, 59], creating risks for users' privacy and security. Unfortunately, organizing a cloud repository by moving files to more sensible locations and deleting unneeded data is cumbersome and time-consuming. Thus, there is a need for tools that help users organize their cloud repositories.

Numerous existing tools [36, 40, 68, 84] 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 [33, 34, 50, 70, 89], but these tools have seen limited adoption, potentially due to users' strong preference for navigating to files through a folder hierarchy [13, 58]. The few tools working over folder hierarchies that do try to help users organize their data, in contexts ranging from cloud repositories to emails [17, 80, 81], 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 [19].

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 intuition that users will want to manage similar files in similar ways. For example, as shown in Figure 1, if 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 *Observation 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. We also extracted ten metadata and content features between each pair of files in participants' repositories. 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 7 in Section 5 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 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.

We first describe related work in Section 2. We present the (shared) methodology of our two user studies in Section 3. We describe the results of the *Observation Study* in Section 4, focusing on findings that guided *KondoCloud*'s design. We detail *KondoCloud*'s design and implementation in Section 5. We present the results of our *Evaluation Study* in Section 6. We conclude with lessons for the design of tools like *KondoCloud* in Section 7.

2 RELATED WORK

KondoCloud builds on prior work related to personal information management, interface support for information management, file search, and recommender systems.

2.1 Personal Information Management

The field of Personal Information Management (*PIM*) has characterized how users acquire [37, 60], keep [57], curate [91], and organize [74–76] their files. The last of these is most relevant to our work. Studies of file organization began in office environments [64, 69], where researchers noted differences between participants who had “neat” versus “messy” desks, identifying potential psychological motivations for these behaviors. Other work identified methods of categorization that people used to organize files. Kwasnik [61–63] conducted several studies among office managers and developed a framework of how office managers categorize files, with attributes such as “use/purpose,” “topic,” “time,” and “value.” Barreau and Nardi [9] carried this stream of work into the digital realm. Boardman and Sasse [19] conducted an empirical study of 31 desktop users, identifying how the categorizations introduced by Kwasnik were used as the basis for folder headings, such as “project,” “document class,” and “role.” Bergman et al. correspondingly put forth the idea that file-management tools should support users' subjective perceptions of file organization in a “user-subjective approach” [12]. These ideas of subjective categorization influence *KondoCloud*'s core idea of recommending file actions.

Given that organization is strongly related to subsequent information retrieval [91], work on file and web page retrieval is also relevant to our work. Prior work has suggested that a large portion

of information consumption consists of “revisitation,” or returning to previously viewed content [4, 26]. Furthermore, users often try to organize their personal storage with later retrieval in mind [91], though not always successfully [19].

Recent research has begun to explore retrieval in cloud storage. Jahanbakhsh et al. [55] investigated users’ recognition and interest in files based on how recently a file was last accessed, as well as the richness of prior interactions. They found some correlation between the recency of access and the time the participant needed to recognize a file. They also found that participants expressed interest in older files despite not having viewed them recently. Xu et al. [95] explored the potential for recommender systems in cloud storage. However, they focused exclusively on recommendations for file retrieval, whereas we focus on actions beyond retrieval, specifically moving and deleting files. Although there have been some long-term studies of file systems [1, 59], few studies look specifically at file accesses. This gap may stem from the difficulty of data collection [22], though novel data-collection methods may improve this situation [32].

2.2 Interfaces for Information Management

A number of researchers have aimed to improve canonical file-and-folder interfaces for browsing files. Gori et al. created 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 [47]. While their tool performs some semi-automated organization, it focuses solely on files with dependencies, whereas KondoCloud makes more general recommendations. KondoCloud’s enhancements to the standard file-browsing interface relate most closely to adaptive interfaces. Greenberg and Witten first identified the potential for interfaces that rearrange in response to user activity [49]. Sears and Schneiderman expanded on this approach by limiting the re-ordering to only occur above a “split” in the menu [79]. Similarly, KondoCloud’s interface consists of a non-adaptive component that resembles standard file browsers, augmented by recommendations that change in response to user activity. Gajos et al. [42] 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 [38, 43]. KondoCloud abides by these principles. BIGFile [68] presents an adaptive interface similar to ours for file retrieval. Unlike KondoCloud, it does not make recommendations beyond file retrieval.

Other work on personal information management has moved beyond the file-and-folder paradigm, proposing alternative interfaces. These interfaces have included Lifestreams’ chronological display of information artifacts [41], Confluence’s time-based contextual retrieval [50], “concept maps” that organize information using a hierarchy of topics [96], and activity-based organization [89]. Some of these tools implicitly help users with disorganization. However, unlike KondoCloud, they require explicit queries or only make



Figure 2: Both the Observation Study and Evaluation Study were conducted in two phases to enable offline processing.

recommendations based on heuristics. Some features of these interfaces have been incorporated into commonly used tools, but these prototypes have found limited adoption, potentially owing to a combination of users’ preference for folders [13, 58] and the difficulty of maintaining research prototypes [30].

2.3 File Search and Recommendation

One potential coping mechanism for finding information within a disorganized repository is for the file interface to provide rich search functionality. Prior work has argued that search has key disadvantages compared to well-organized repositories. Teevan et al. [85], Bergman et al. [13], and Bergman et al. [14] conducted studies on this topic using semi-structured interviews, longitudinal measurement, and an in-lab study. They found that search has a higher cognitive burden than navigating through a file-browser interface, and that forming a search query requires a user to recall some context for the file without any aid. Teevan et al. found that users navigate through the file hierarchy by using additional context gained at each step of navigation to orient themselves toward their goal [85]. Some recent work has sought to overcome these burdens, such as by using chatbots during searches [5] to reduce cognitive load. KondoCloud avoids these issues by offering recommendations from users’ behavior without requiring explicit queries.

Researchers have also proposed recommender systems for file retrieval. Based on the idea of predicting a user’s next file access [39], these tools may highlight files or folders of interest [40], provide shortcuts to files of interest [68], or work on top of common tools like Google Drive [23, 84]. Some tools, such as GrayArea [15] and Favorite Folders [65], give users manual control over de-emphasizing less-accessed files and folders. KondoCloud improves on these approaches by recommending actions beyond retrieval. Several tools [17, 80, 81] similarly save documents or emails to suggested folders, but only examine the initial save point of information. Further, these and similar tools use access patterns as the key feature for making predictions, whereas KondoCloud uses a much richer set of ten different file metadata and content features.

3 METHODOLOGY

We conducted two online user studies, the Observation Study and the Evaluation Study. The studies followed similar protocols (see Figure 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

Table 1: The similarity features we collected in Part 1 of each study. Each feature was computed pairwise for files in participants’ repositories. This approach builds on Brackenburg et al. [21]. Metadata features were calculated between all file pairs, while text and image features were only calculated when both files in a pair were text or images, respectively.

Feature	Description
Metadata	
<i>Last Modified</i>	Logarithm of difference, in seconds, between the two files’ last modified dates
<i>Filename</i>	Jaccard similarity of the list of bigrams (two-letter chunks) in the filenames
<i>File Size</i>	Logarithm of difference, in bytes, between the file size
<i>Tree Distance</i>	The number of steps to reach one file from the other when traversing the file hierarchy (represented as a tree)
<i>Shared Users</i>	Jaccard similarity of the lists of unique user IDs with whom the files have been shared
<i>File Contents</i>	Jaccard similarity of chunks of raw file content using MinHash
Text	
<i>Text Contents</i>	Cosine similarity between documents’ Word2Vec [72] vector embeddings
<i>Text Topic</i>	Cosine similarity of documents’ Term Frequency Inverse Document Frequency (TF-IDF) vectors [94]
Images	
<i>Image Contents</i>	Jaccard similarity between unique objects recognized in images by object-detection algorithms [46]
<i>Image Color</i>	Absolute difference between the average RGB values for each photo

than local storage [22], 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 [29].

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. Our code subsequently began extracting ten types of file metadata and content features (see Table 1) for pairs of files in their repository [21]. 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.

3.1 Recruitment and Part 1

We recruited participants on the Prolific crowdsourcing marketplace [77]. 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 [52], and the names of

columns in spreadsheets. We further computed the 10 metadata and content similarity features described in Table 1 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 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.

3.2 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* [83]. 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 7 in Section 5. 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 preliminary data collected in our previous studies [21]. In Evaluation Study, we used the classifier as described in Section 5. 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 [86]. 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.

3.3 Ethics

Our protocols were approved by the University of Chicago’s Institutional Review Board (IRB). Since our use of the Google Drive API would fall outside Prolific’s permitted use cases, we reached out to Prolific’s customer support before beginning the study, and they granted us an exception to their policies. Our app for interfacing with Google Drive was also approved by Google as part of their standard review process for apps. Because some participants were from the UK, our data collection and processing procedures also followed all relevant UK-GDPR requirements. To protect participant privacy, we did not store their raw files, but instead only

extracted the features (and in many cases only the difference in features between pairs of files) necessary to train our classifier.

3.4 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 [88]. 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 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 [19, 27, 31] or asked users to describe, abstractly and qualitatively, how they organize [8, 69, 75, 76]. However, repositories typically do not become more organized over time, and qualitative studies of organization may miss fine-grained strategies.

4.1 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 2 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”

Table 2: Characteristics of participants' Google Drive repositories prior to organization.

	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

with a particularly large number of files is consistent with prior work on local file systems [31]. 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 that noted that file repositories generally follow patterns of either storing files in a small number of folders (called “pilers,” “one-folder filing,” “hoarding,” or “fuzzy”) or in a rich and complex folder hierarchy (“filers,” “total filers,” “structurers,” or “rigid”) [19, 51, 54, 69, 75, 87]. 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 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.

4.2 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 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 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 participant who only moved files and folders labeled their clusters of actions as “backup documents” and “my personal files.” 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 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 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 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 5 shows the relative frequency of pairs of file actions. Three patterns are evident. Participants often grouped folder

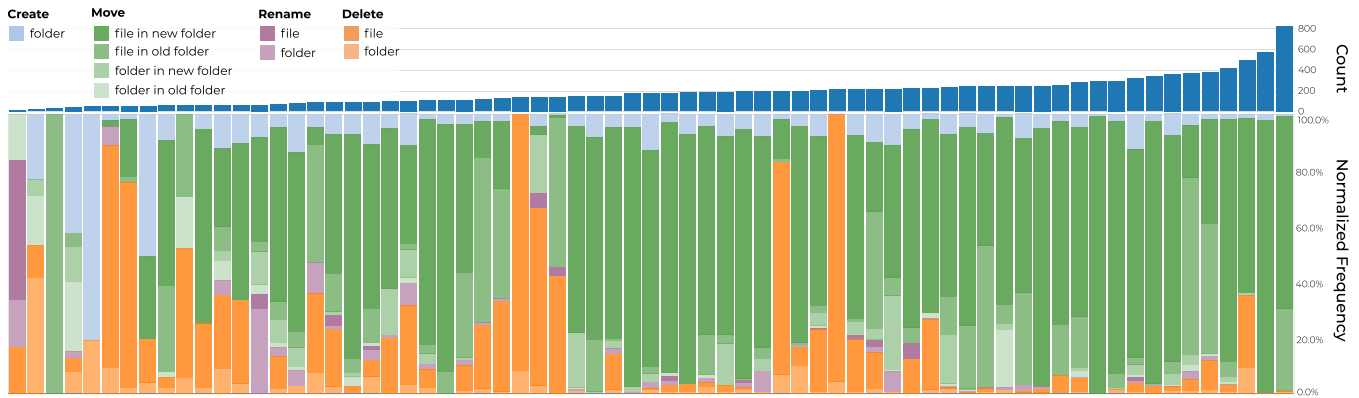


Figure 3: Participants’ file-management actions while organizing their Google Drive repository in the Observation Study. The x-axis is ordered by the total number of file-management actions the participant took, which is also shown in the bar graph (top). The stacked-bar graph (bottom) shows the distribution of different types of actions.

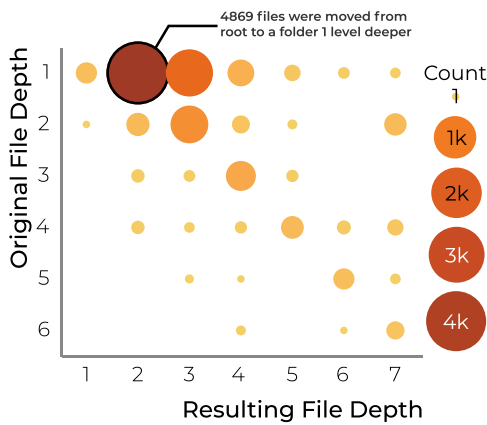


Figure 4: Files were typically moved deeper into the file hierarchy regardless of file type.

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’s idea of task-based context [7], 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. While KondoCloud’s straightforward recommendations for individual file actions do not yet capture our observation that file creation events were typically followed by moving files or folders into that newly created folder, future tools might consider suggesting the creation of a new folder and bulk moves of related files to that folder.

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 6 shows the number of file actions performed during different temporal segments of the organization task. The mean number of move and delete actions during any normalized time block did not differ substantially. This finding suggested a balanced approach for

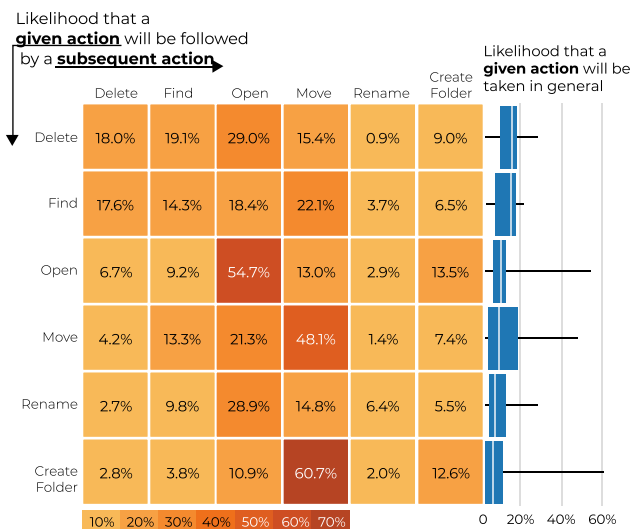


Figure 5: Probability of actions following others. Participants often followed moving or opening files with other moves. They also often followed folder creation with moves.

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.

5 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 [17, 36, 40, 68, 80, 81, 84], but KondoCloud is the first to offer recommendations that retrospectively address existing disorganization

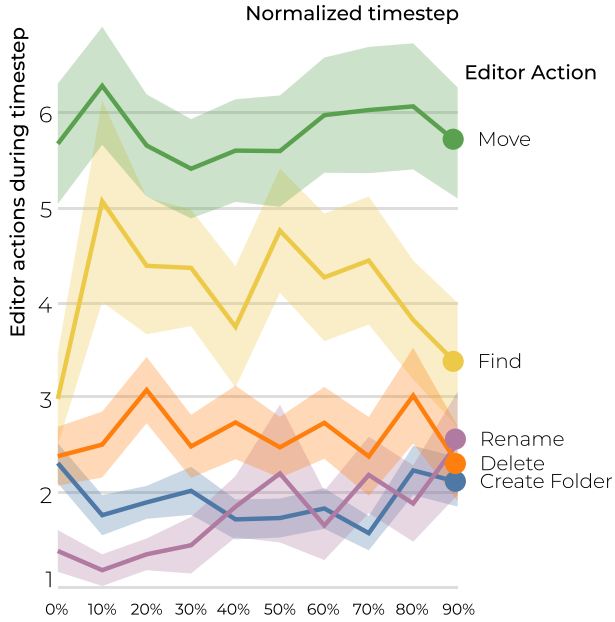


Figure 6: Distribution of actions taken (across all participants) during the ten normalized time steps.

in cloud repositories. We have released KondoCloud’s code open source on GitHub.¹

5.1 Interface Design

The basis for the KondoCloud interface is a standard file browser, shown on the right side of Figure 7. Starting from an open source file-browser interface, elFinder [83], 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” [10, 24]. 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 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 [10]. Hovering over a recommendation card explains the recommendation by showing the file action that triggered the recommendation, as in prior work [95]. 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 [42, 68, 79], 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 [42]. We analyze in the Evaluation Study the degree to which participants directly accepted recommendations, versus performing the recommended actions manually in the file browser.

5.2 Recommendations

KondoCloud generates recommendations as follows. Each time the user moves, deletes, or previews a file, we use our machine-learning classifier (see Section 5.3) 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 [53]. We hypothesized that providing a shortcut to perform the action would enhance file organization in a number of ways (see Section 6).

More precisely, we choose whether to show a recommendation using a probability threshold as shown in Algorithms 1–2:

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(
       initial_change, decay_factor, move_number_of_updates)
7:   delete_change_value ← CalculateValueChange(
       initial_change, decay_factor, delete_number_of_updates)
8:   if participant rejects recommendation then
9:     if recommendation type is move then
10:      move_number_of_updates += 1
11:      move_threshold += move_change_value
12:     else if recommendation type is delete then
13:      delete_number_of_updates += 1
14:      delete_threshold += delete_change_value
15:   end if
16:   else if participant accepts recommendation then
17:     if recommendation type is move then
18:      move_number_of_updates += 1
19:      move_threshold -= move_change_value
20:     else if recommendation type is delete then
21:      delete_number_of_updates += 1
22:      delete_threshold -= delete_change_value
23:   end if
24:   end if
25: end while

```

¹<https://github.com/UChicagoSUPERgroup/kondocloud>

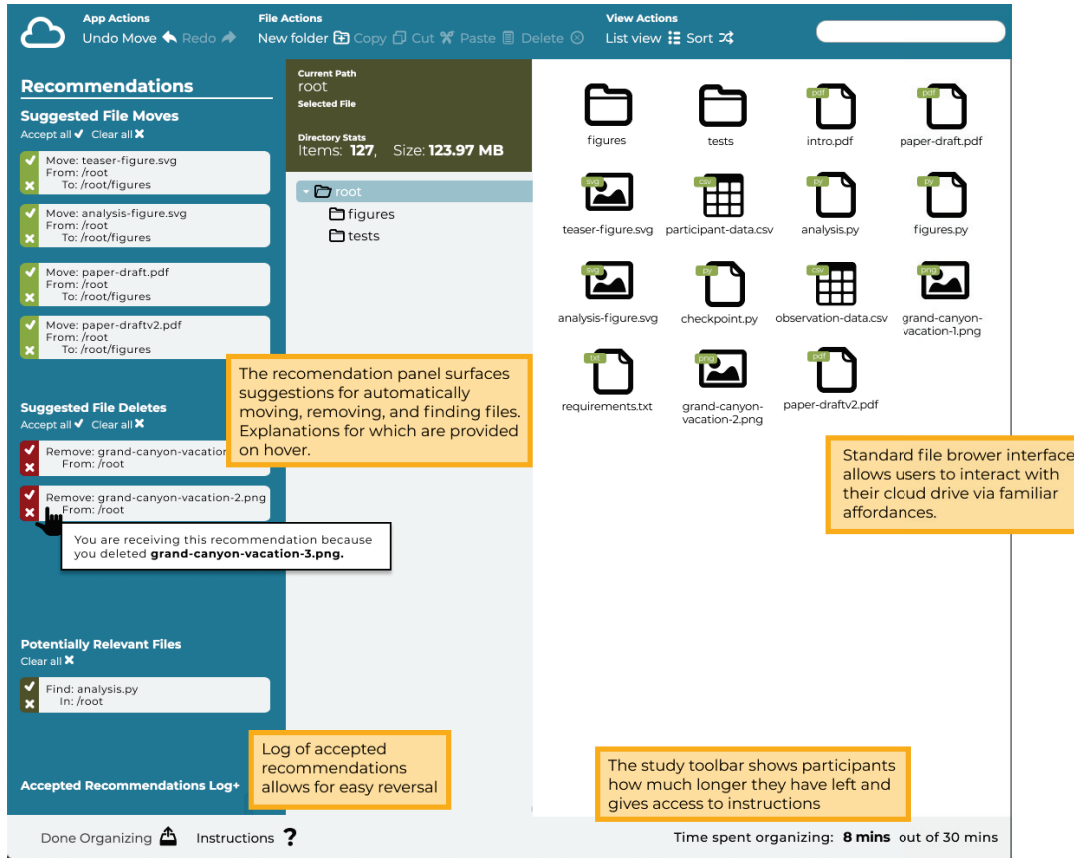


Figure 7: The KondoCloud interface augments a traditional file browser with context-dependent, ML-based recommendations.

Algorithm 2 CalculateValueChange

```

1: Takes arguments initial_change, decay_factor,
   number_of_updates
2: Initialize update,  $\leftarrow$  initial_change
3: for all  $i \in \{1..number\_of\_updates\}$  do
4:    $update \leftarrow power(update, 1 + decay\_factor)$ 
5: end for
6: return update

```

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, 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 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 and Figure 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.

In keeping with principles identified in prior work, recommendations are easily dismissed or corrected [3]. 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 [93]. 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 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 [7]. However, per Figure 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.

5.3 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 3.2, 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 described in Table 1 (Section 3.1) 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 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 [42, 66].

6 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.

6.1 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

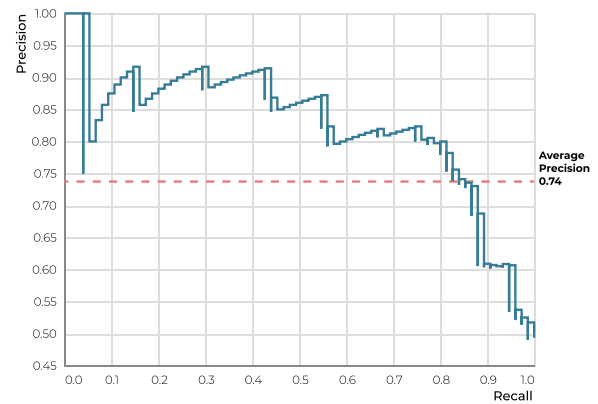


Figure 8: Precision-recall curve for the overall classifier.

Table 3: Coefficients (β) of the three Logistic Regression classifiers we created. Our overall classifier (Figure 8) chooses the appropriate model based on the types (text, image, or other) of the two files being compared.

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

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. Participants again varied in their organizational strategies and actions.

6.2 Outcome of Recommendations

KondoCloud’s recommendations formed a core component of many participants’ organizational workflows. Figure 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%

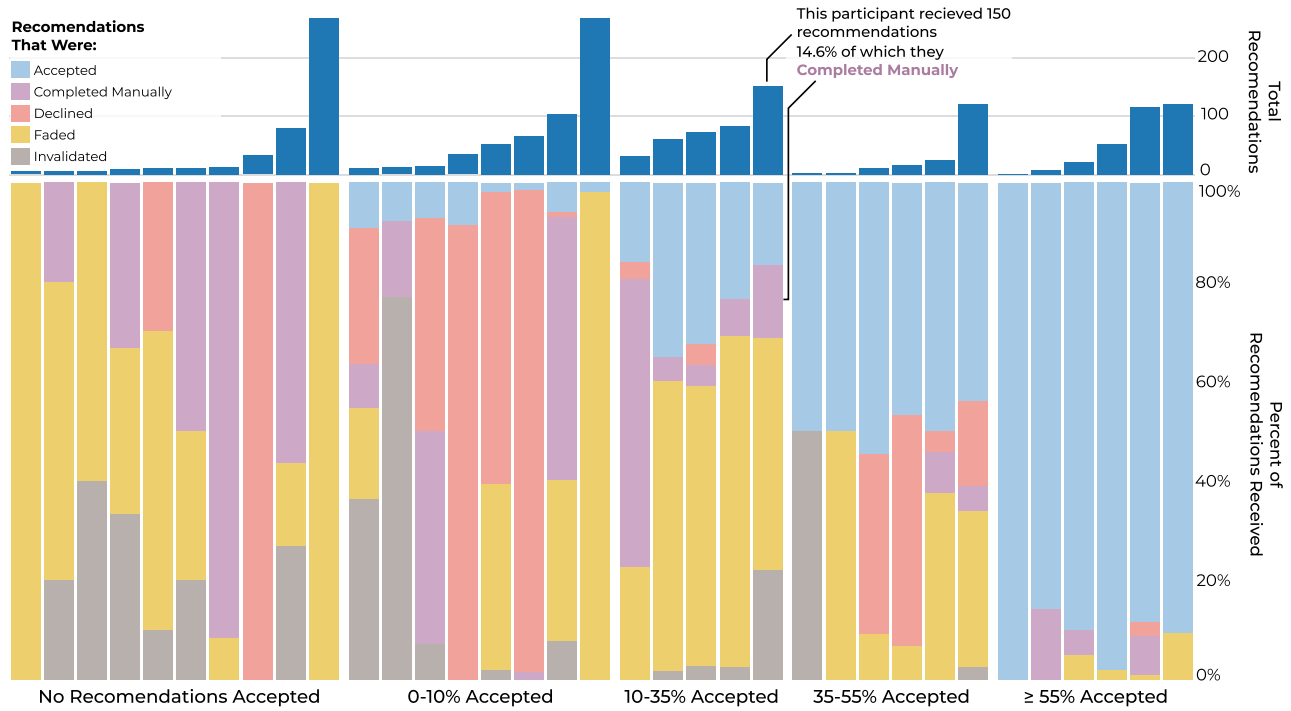


Figure 9: The number of recommendations generated for each participant based on their organizational actions (top), as well as the outcome of those recommendations (bottom). We cluster participants on the fraction of recommendations accepted.

(5, 13.9%), 0-10% (8, 22.2%) and those who accepted none (11, 30.6%). As seen in the top portion of Figure 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 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.

6.3 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 recommendations and asked participants to respond to a statement that the recommendation improved the efficiency of organizing their account (Figure 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 [73]. 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.

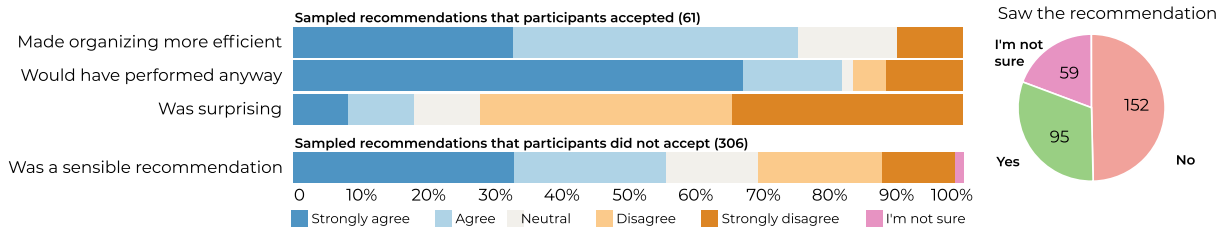


Figure 10: Participants' responses to questions about a sample of 61 recommendations they accepted (left, top), 306 recommendations they did not accept (left, bottom), and whether they remembered seeing specific recommendations (right).

As seen in Figure 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 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 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 [59], 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 11. This finding is not surprising because most moves are from the root to a subfolder one level below (see Figure 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 12 shows that 33.3% of participants reported finding recommendations useful, 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 [6]. We did not observe significant differences in SUS scores across conditions or relative to the proportion of recommendations a participant accepted.

KondoCloud is a step toward helping participants organize, find, and delete files, yet can be improved in several directions. First, 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. It may be useful to present these related recommendations as a group, allowing the user to accept all, reject all, or view the individual recommendations on demand. In fact, 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,” explaining that it would be faster or easier. An open challenge, though, is how to communicate what actions are contained in a group.

Second, KondoCloud’s current design does not take into account information about the current file hierarchy. As seen in Figure 4, during the Observation Study, organization actions almost always

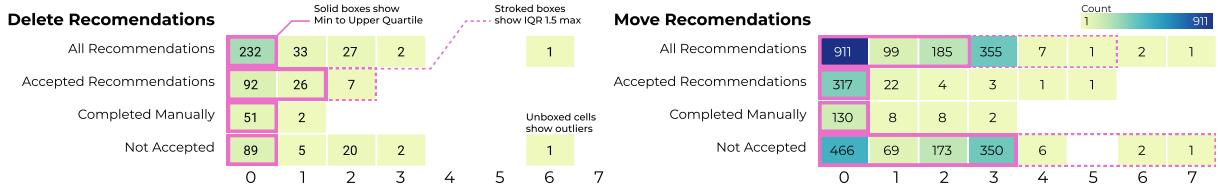


Figure 11: Distribution of tree distance between recommended file pairs and the outcome of the recommendation, shown as a heatmap with a boxplot encoded as the borders of the boxes.

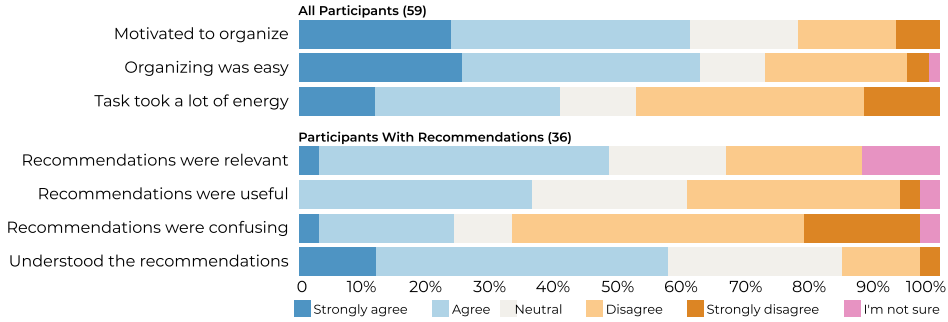


Figure 12: Responses to questions about the general organization task (top) and recommendations (bottom).

increased the depth of a file or folder’s placement. However, the classifier underlying KondoCloud currently only estimates the quality of a recommendation based on the similarity between files, ignoring information about the location to which a file would be moved. However, one result of this intentional choice was that 683 (43.8%) of the 1,561 move recommendations made in the With Recommendations condition in the Evaluation Study recommended negative changes in file depth (e.g., moving a file several subfolders deep to a folder closer to the root). Only 2 such recommendations were accepted, and only 13 were completed manually, showing that the suggested file depth change is a significant factor in whether a recommendation is accepted (Spearman’s rank correlation, p -value < 0.001). As a result, although the machine-learning model underlying KondoCloud focuses purely on file similarity, effectively leveraging some information about the proposed action (e.g., destination folder) could enhance recommendations noticeably.

7 DISCUSSION AND CONCLUSION

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.

We envision several ways future work could improve KondoCloud. First, *grouping recommendations* may make it easier to accept recommendations for sets of many related actions. For example, if a user moved one vacation photo from the root folder to a new sub-folder and the system identified several related photos, the resulting recommendations could be grouped under a single heading, with the ability to accept or dismiss all of them with one click.

Second, *modifying our pre-processing approach to improve scalability* would improve KondoCloud’s ability to handle large repositories. Because our 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 [45], learned hashing methods [90], or quantization methods [44, 48, 56] 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 [67, 71, 82]. 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. [2] 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 [28]. Further, incorporating early user feedback via more intrusive notifications, such as negotiated-style interruptions [78] at the start of organizing could enable KondoCloud to personalize recommendations quickly.

Generalizing KondoCloud to other domains requires additional research. KondoCloud is designed on top of a finder-style interface [40], which differs from some current cloud storage interfaces (e.g., Google Drive). The ability to extend KondoCloud to such settings depends on how well such interfaces implement the guarantees of finder-style interfaces, such as folder-based organization and files having a fixed location [24, 34]. Extending KondoCloud to local storage may also require further studies of the differences in co-management behavior in non-shared environments [88].

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants No. CNS-1801663 and OAC-1835890, as well as the CERES Center for Unstoppable Computing. We thank our reviewers for their thoughtful comments and suggestions.

REFERENCES

- [1] Nitin Agrawal, William J. Bolosky, John R. Douceur, and Jacob R. Lorch. 2007. A five-year study of file-system metadata. *ACM TOS* 3, 3 (2007), 9.
- [2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.
- [3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proc. CHI*.
- [4] Anne Aula, Natalie Jhaveri, and Mika Käki. 2005. Information search and re-access strategies of experienced web users. In *Proc. WWW*.
- [5] Sandeep Avula, Gordon Chadwick, Jaime Arguello, and Robert Capra. 2018. SearchBots: User engagement with chatbots during collaborative search. In *Proc. CHIIR*.
- [6] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies* 4, 3 (2009), 114–123.
- [7] Xinlong Bao and Thomas G. Dietterich. 2011. FolderPredictor: Reducing the cost of reaching the right folder. *ACM TIST* 2, 1 (2011).
- [8] Deborah Barreau. 1995. Context as a factor in personal information management systems. *Journal of the American Society for Information Science* 46, 5 (1995), 327–339.
- [9] Deborah Barreau and Bonnie A. Nardi. 1995. Finding and reminding: File organization from the desktop. *ACM SIGCHI Bulletin* 27, 3 (1995), 39–43.
- [10] Yael Bann, Ofer Bergman, Liv Glazer, Paris Arent, Iain D. Wilkinson, Rosemary Varley, and Steve Whittaker. 2015. Navigating through digital folders uses the same brain structures as real world navigation. *Scientific Reports* 5, 1 (2015).
- [11] Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. 2006. The project fragmentation problem in personal information management. In *Proc. CHI*.
- [12] Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. 2008. The user-subjective approach to personal information management systems design: Evidence and implementations. *JASIST* 59, 2 (2008), 235–246.
- [13] Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. 2008. Improved search engines and navigation preference in personal information management. *ACM TOIS* 26, 4 (2008).
- [14] Ofer Bergman, Maskit Tene-Rubinstein, and Jonathan Shalom. 2013. The use of attention resources in navigation versus search. *Personal and Ubiquitous Computing* 17, 3 (2013), 583–590.
- [15] Ofer Bergman, Simon Tucker, Ruth Beyth-Marom, Edward Cutrell, and Steve Whittaker. 2009. It's not that important: Demoting personal information of low subjective importance using GrayArea. In *Proc. CHI*.
- [16] Ofer Bergman, Steve Whittaker, and Noa Falk. 2014. Shared files: The retrieval perspective. *JASIST* 65, 10 (2014), 1949–1963.
- [17] Ofer Bergman, Steve Whittaker, and Yaron Frishman. 2019. Let's get personal: The little nudge that improves document retrieval in the cloud. *J. Doc* (2019).
- [18] Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. 2010. The effect of folder structure on personal file navigation. *JASIST* 61, 12 (2010), 2426–2441.
- [19] Richard Boardman and M. Angela Sasse. 2004. Stuff goes into the computer and doesn't come out: A cross-tool study of personal information management. In *Proc. CHI*.
- [20] Richard Boardman, Robert Spence, and M. Angela Sasse. 2003. Too many hierarchies? The daily struggle for control of the workspace. In *Proc. HCI*.
- [21] Will Brackenburg, Galen Harrison, Kyle Chard, Aaron Elmore, and Blase Ur. 2021. Files of a feather flock together? Measuring and modeling how users perceive file similarity in cloud storage. In *Proc. SIGIR*.
- [22] Robert Capra and Manuel A. Pérez-Quinones. 2006. Factors and evaluation of refinding behaviors. In *Proc. SIGIR*.
- [23] 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. 2020. Improving recommendation quality at Google Drive. In *Proc. KDD*.
- [24] Andrea Civan, William Jones, Predrag Klasnja, and Harry Bruce. 2008. 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), 1–13.
- [25] Jason W. Clark, Peter Snyder, Damon McCoy, and Chris Kanich. 2015. I saw images I didn't even know I had: Understanding user perceptions of cloud storage privacy. In *Proc. CHI*.
- [26] Andy Cockburn and Bruce McKenzie. 2001. What do web users do? An empirical analysis of web use. *IJHCS* 54, 6 (2001), 903–922.
- [27] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A diary study of task switching and interruptions. In *Proc. CHI*.
- [28] Sanjoy Dasgupta and Sivan Sabato. 2020. Robust learning from discriminative feature feedback. In *Proc. AISTATS*.
- [29] Jesse David Dinneen and Ilja Frissen. 2020. Mac users do it differently: The role of operating system and individual differences in file management. In *Proc. CHI Extended Abstracts*.
- [30] Jesse David Dinneen and Charles-Antoine Julien. 2015. The disappearing semantic web; an examination of 54 semantic search tools. In *Proceedings of the Annual Conference of CAIS*.
- [31] Jesse David Dinneen, Charles-Antoine Julien, and Ilja Frissen. 2019. The scale and structure of personal file collections. In *Proc. CHI*.
- [32] Jesse David Dinneen, Fabian Odoni, Ilja Frissen, and Charles-Antoine Julien. 2016. Cardinal: Novel software for studying file management behavior. In *Proceedings of the 79th ASIS&T Annual Meeting*.
- [33] Paul Dourish. 2003. The appropriation of interactive technologies: Some lessons from placeless documents. *CSCW* 12, 4 (2003), 465–490.
- [34] Paul Dourish, W. Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen, Michael Salisbury, Douglas B. Terry, and James Thornton. 2000. Extending document management systems with user-specific active properties. *ACM TOIS* 18, 2 (2000), 140–170.
- [35] Idilio Drago, Marco Mellia, Maurizio M. Munafò, Anna Sperotto, Ramin Sadre, and Aiko Pras. 2012. Inside Dropbox: Understanding personal cloud storage services. In *Proc. IMC*.
- [36] Susan Dumais, Edward Cutrell, Jonathan J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've seen: A system for personal information retrieval and re-use. In *Proc. SIGIR*.
- [37] David Ellis and Merete Haugan. 1997. Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *J. Doc* 53, 4 (1997), 384–403.
- [38] Leah Findlater and Joanna McGrenere. 2010. Beyond performance: Feature awareness in personalized interfaces. *IJHCS* 68, 3 (2010), 121–137.
- [39] Stephen Fitchett and Andy Cockburn. 2012. Accessrank: Predicting what users will do next. In *Proc. CHI*.
- [40] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. 2014. Finder highlights: Field evaluation and design of an augmented file browser. In *Proc. CHI*.
- [41] Eric Freeman and David Gelernter. 1996. Lifestreams: A storage model for personal data. *ACM SIGMOD Record* 25, 1 (1996), 80–86.
- [42] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI*.
- [43] Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. 2008. Predictability and accuracy in adaptive user interfaces. In *Proc. CHI Extended Abstracts*.
- [44] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization for approximate nearest neighbor search. In *Proc. CVPR*.
- [45] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity search in high dimensions via hashing. In *Proc. VLDB*.
- [46] Google. 2019. Vision AI. <https://cloud.google.com/vision/>.
- [47] Julien Gori, Han L. Han, and Michel Beaudouin-Lafon. 2020. FileWeaver: Flexible file management with automatic dependency tracking. In *Proc. UIST*.
- [48] Robert Gray. 1984. Vector quantization. *IEEE ASSP Magazine* 1, 2 (1984), 4–29.
- [49] Saul Greenberg and Ian H. Witten. 1985. Adaptive personalized interfaces—A question of viability. *Behaviour & Information Technology* 4, 1 (1985), 31–45.
- [50] Karl Gyllstrom. 2009. *Enriching personal information management with document interaction histories*. Ph.D. Dissertation.

- [51] Sharon Hardof-Jaffe, Arnon Hershkovitz, Hama Abu-Kishk, Ofer Bergman, and Rafi Nachmias. 2009. Students' Organization Strategies of Personal Information Space. *Journal of Digital Information* 10, 5 (2009).
- [52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*.
- [53] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *PNAS* 116, 6 (2019), 1844–1850.
- [54] Sarah Henderson and Ananth Srinivasan. 2009. An empirical analysis of personal digital document structures. In *Proc. HCI*.
- [55] Farnaz Jahanbakhsh, Ahmed Hassan Awadallah, Susan T. Dumais, and Xuhai Xu. 2020. Effects of past interactions on user experience with recommended documents. In *Proc. CHIIR*.
- [56] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE TPAMI* 33, 1 (2010), 117–128.
- [57] William Jones, Susan Dumais, and Harry Bruce. 2002. 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 (2002), 391–402.
- [58] William Jones, Ammy Jiranida Phuwanturak, Rajdeep Gill, and Harry Bruce. 2005. Don't take my folders away! Organizing personal information to get things done. In *Proc. CHI Extended Abstracts*.
- [59] Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. 2018. Forgotten but not gone: Identifying the need for longitudinal data management in cloud storage. In *Proc. CHI*.
- [60] Carol C. Kuhlthau. 1991. Inside the search process: Information seeking from the user's perspective. *JASIST* 42, 5 (1991), 361–371.
- [61] Barbara H. Kwasnik. 1989. How a personal document's intended use or purpose affects its classification in an office. In *ACM SIGIR Forum*, Vol. 23. 207–210.
- [62] Barbara H. Kwasnik. 1991. The importance of factors that are not document attributes in the organization of personal documents. *J. Doc* (1991).
- [63] Barbara H. Kwasnik. 1992. The role of classification structures in reflecting and building theory. *Advances in Classification Research Online* 3, 1 (1992), 63–82.
- [64] Mark W. Lansdale. 1988. The psychology of personal information management. *Applied Ergonomics* 19, 1 (1988), 55–66.
- [65] Bongshin Lee and Benjamin B. Bederson. 2003. *Favorite folders: A configurable, scalable file browser*. Technical Report.
- [66] John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors* 46, 1 (2004), 50–80.
- [67] Jiahui Liu, Peter Dolan, and Elin Ronby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proc. IUI*.
- [68] Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile: Bayesian information gain for fast file retrieval. In *Proc. CHI*.
- [69] Thomas W. Malone. 1983. How do people organize their desks?: Implications for the design of office information systems. *ACM TOIS* 1, 1 (1983), 99–112.
- [70] Gary Marsden and David E. Cairns. 2004. Improving the usability of the hierarchical file system. *South African Computer Journal* 2004, 32 (2004), 69–78.
- [71] Ben McCamish, Vahid Ghadakhchi, Arash Termehchy, Behrouz Touri, and Liang Huang. 2018. The data interaction game. In *Proc. SIGMOD*.
- [72] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*.
- [73] Xi Niu and Ahmad Al-Doulat. 2021. LuckyFind: Leveraging surprise to improve user satisfaction and inspire curiosity in a recommender system. In *Proc. CHIIR*.
- [74] Kyong Eun Oh. 2012. What happens once you categorize files into folders? *Proceedings of the American Society for Information Science and Technology* 49, 1 (2012), 1–4.
- [75] Kyong Eun Oh. 2017. Types of personal information categorization: Rigid, fuzzy, and flexible. *JASIST* 68, 6 (2017), 1491–1504.
- [76] Kyong Eun Oh and Nicholas J. Belkin. 2014. Understanding what personal information items make categorization difficult. *Proceedings of the American Society for Information Science and Technology* 51, 1 (2014), 1–3.
- [77] Prolific. 2019. <https://www.prolific.co/>.
- [78] T.J. Robertson, Shrinu Prabhakararao, Margaret Burnett, Curtis Cook, Joseph R. Ruthruff, Laura Beckwith, and Amit Phalgune. 2004. Impact of interruption style on end-user debugging. In *Proc. CHI*.
- [79] Andrew Sears and Ben Shneiderman. 1994. Split menus: effectively using selection frequency to organize menus. *ACM TOCHI* 1, 1 (1994), 27–51.
- [80] Richard B. Segal and Jeffrey O. Kephart. 1999. MailCat: An intelligent assistant for organizing e-mail. In *Proc. AGENTS*.
- [81] Debmalya Sinha and Anupam Basu. 2012. Gardener: A file browser assistant to help users maintaining semantic folder hierarchy. In *Proc. IHCI*.
- [82] Aleksandrs Slivkins. 2011. Contextual bandits with similarity information. In *Proc. COLT*.
- [83] Studio 42. 2019. <https://studio-42.github.io/elFinder/>.
- [84] Sandeep Tata, Alexandrin Popescu, Marc Najork, Mike Colagrosso, Julian Gibbons, Alan Green, Alexandre Mah, Michael Smith, Divanshu Garg, Cayden Meyer, and Reuben Kan. 2017. Quick access: building a smart experience for Google Drive. In *Proc. KDD*.
- [85] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 2004. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proc. CHI*.
- [86] Usability.gov. 2021. <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
- [87] Francesco Vitale, Isabelle Janzen, and Joanna McGrenere. 2018. Hoarding and minimalism: Tendencies in digital data preservation. In *Proc. CHI*.
- [88] Amy Volda, Judith S. Olson, and Gary M. Olson. 2013. Turbulence in the clouds: Challenges of cloud-based information work. In *Proc. CHI*.
- [89] Stephen Volda and Elizabeth D. Mynatt. 2009. It feels better than filing: Everyday work experiences in an activity-based computing system. In *Proc. CHI*.
- [90] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. 2015. Learning to hash for indexing big data—A survey. *Proc. IEEE* 104, 1 (2015), 34–57.
- [91] Steve Whittaker. 2011. Personal information management: From information consumption to curation. *Annual Review of Information Science and Technology* 45, 1 (2011), 1–62.
- [92] Steve Whittaker, Ofer Bergman, and Paul Clough. 2010. Easy on that trigger dad: A study of long term family photo retrieval. *Personal and Ubiquitous Computing* 14, 1 (2010), 31–43.
- [93] Brian Whitworth. 2005. Polite computing. *Behaviour & Information Technology* 24, 5 (2005), 353–363.
- [94] Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, and Kui-Lam Kwok. 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM TOIS* 26, 3 (2008).
- [95] Xuhai Xu, Ahmed Hassan Awadallah, Susan T. Dumais, Farheen Omar, Bogdan Popp, Robert Rounthwaite, and Farnaz Jahanbakhsh. 2020. Understanding user behavior for document recommendation. In *Proc. WWW*.
- [96] Liang Huai Yang, Jian Zhou, Jiacheng Wang, and Mong-Li Lee. 2012. A novel PIM system and its effective storage compression scheme. *JSW* 7, 6 (2012), 1385–1392.