

Orchard: Exploring Multivariate Heterogeneous Networks on Mobile Phones

P. Eichmann¹, D. Edge², N. Evans³, B. Lee³, M. Brehmer^{4†}, and C. White³

¹Brown University, Providence, RI, USA ²Microsoft Research, Cambridge, UK

³Microsoft Research, Redmond, WA, USA ⁴Tableau Research, Seattle, WA, USA

Abstract

People are becoming increasingly sophisticated in their ability to navigate information spaces using search, hyperlinks, and visualization. But, mobile phones preclude the use of multiple coordinated views that have proven effective in the desktop environment (e.g., for business intelligence or visual analytics). In this work, we propose to model information as multivariate heterogeneous networks to enable greater analytic expression for a range of sensemaking tasks while suggesting a new, list-based paradigm with gestural navigation of structured information spaces on mobile phones. We also present a mobile application, called Orchard, which combines ideas from both faceted search and interactive network exploration in a visual query language to allow users to collect facets of interest during exploratory navigation. Our study showed that users could collect and combine these facets with Orchard, specifying network queries and projections that would only have been possible previously using complex data tools or custom data science.

CCS Concepts

• **Human-centered computing** → **Information visualization; Visualization systems and tools; User interface design; Information systems** → **Search interfaces;**

1. Introduction

People regularly interact with complex information networks that model the attributes and relationships of real-world entities. Whether through browsing hyperlinked networks or the ranked lists of search results, navigating between related pages, people, products, and media is a universal quality of the online experience. At the same time as the search, list, and link paradigm is expanding to ever more information seeking experiences, the trend towards integrating analytic sensemaking capabilities in consumer experiences such as video games and fantasy sports is driving the adoption of business-like dashboards beyond the business context [Mee17].

The representation and analysis of complex information networks has rich history in the fields of information visualization and visual analytics, with the analysis of social networks (e.g., [BCD*10]) and citation networks (e.g., [LCRB05, KPLB07, CKHF11, ZCCB13]) featuring prominently. Additional use cases include exploratory analysis of data published by governments (e.g., migration flows [VdEVW14]) or businesses (e.g., company-investor networks [CKHF11]), as well as investigative analysis of potential impropriety or illegality (e.g., in Enron email communications [SA04], Panama Papers documentation [Lea19], or “open secrets” interactions between politicians and lobbyists [ope19]).

What is less well examined in the literature, however, is how such information networks may be explored on mobile devices. We posit that there are potentially significant latent demands across many everyday activities increasingly performed on mobile phones. These include sports and gaming analysis (including fantasy sports and e-sports), media consumption (e.g., music, books, films and videos, news), and personal research into major life decisions (e.g., jobs, houses, schools, cars, vacations). In this work, we present a new interaction paradigm for exploring such information on mobile phones.

We draw design inspiration from faceted approaches to search, browsing, and analysis in which users select from prefabricated filters. These filters are automatically extracted from multiple orthogonal data dimensions and their application has the effect of progressively reducing the number of matching entities to a meaningful and manageable subset. Notable examples of this approach from the literature include FacetLens [LSR*09] and Immense [LJH13], which present multiple coordinated views in a fixed 2D layout, and GraphTrail [DHRL*12] and PanoramicData [ZZD14] that provide freeform 2D canvases on which to compose queries and inspect results. Business Intelligence dashboards in general-purpose tools like Tableau, Qlik, and Power BI may also be constructed in a faceted style (e.g., to allow browsing of news and social media “documents” [ELW18]).

† Matt conducted this work while with Microsoft Research.

We note that all such faceted interfaces have deficiencies when applied to multivariate heterogeneous networks on mobile phones. First, faceted interfaces do not convey the underlying heterogeneous network structure. They do not present how facets of nodes in a network or nodes themselves relate to one other, making it difficult for users to grasp the concept of pivoting between different node types. Second, faceted interfaces as developed in the literature do not translate well to mobile phones, where space is limited. While it is technically feasible to employ such interfaces within mobile applications, they would likely impair the user experience. For example, navigating between views in 2D layouts would require users to pan and zoom excessively (analogous to when browsing non-mobile-optimized websites).

To address these issues, we present Orchard, a solution that applies faceted search concepts to the exploratory analysis of multivariate heterogeneous networks on mobile phones. Our design draws on the familiar concept of keyword search and result listing, a paradigm that has already established itself as the dominant form of mobile information seeking. With Orchard, users can explore aggregated node and edge attributes through sortable list-based views that simultaneously function as horizontal bar charts for aggregate values. Orchard combines faceted search on these views with lateral pivoting (following links across one or many hops) that automatically “collects” the ensuing facet trail for future use.

Orchard pivoting extends prior work on interactive network exploration by allowing free navigation between nodes of *any* type, whether or not they are connected directly in the data model. By inferring a full graph query specification from any partial query, demand for additional user input is minimized and the result is an uninterrupted pivoting experience. Orchard also introduces the notion of *reflective pivoting*, i.e., pivoting from/to the same node type. While reflective pivoting is usually considered a *no-op*, Orchard fills this semantic gap by interpreting commands to pivot nodes onto themselves as a command to project the nodes into homogeneous network in which nodes are connected based on shared connections to nodes of another type (as selected by users). This allows users to seamlessly transform and view any subset of a heterogeneous network as a homogeneous network, while collecting more expressive facets in the process (such as pairs of nodes connected by highly-weighted links).

In summary, our contributions are:

- We introduce and motivate the concept of interacting with multivariate heterogeneous networks on mobile phones.
- We present a novel query specification model for multivariate heterogeneous networks called *Pivot Trails* that is particularly suitable to be implemented on small screen devices, requiring minimal user input to formulate network queries.
- We design and implement *Orchard*, a mobile application that uses *Pivot Trails* to support casual exploratory workflows over multivariate heterogeneous networks using a list-based paradigm with gestural navigation.
- We report a qualitative evaluation of the design of Orchard with 12 participants. Initial results show that users quickly understand Orchard’s data and query model, and that they are able to accomplish common exploratory network tasks.

2. Background and Related Work

Orchard relates to prior work on supporting information exploration in heterogeneous information networks, and is inspired by work on exploratory search [Mar06, WR09]. An information network represents an abstraction of the real world, focusing on the objects (nodes) and their attributes, and interactions (links) between the objects [SH12]. In multivariate heterogeneous networks, multivariate refers to the fact that each node or link in a network can have multiple attributes, and heterogeneous indicates that there are multiple node or link types in a network.

2.1. Core Concepts

There is a large body of prior work on facilitating exploration and analysis of multivariate heterogeneous networks, including the development of taxonomies describing the nature of tasks associated with such networks [LPP*06, APS13, PPS14]. We have identified five core concepts (CC1-5) that capture the characteristics of prior network exploration systems.

CC1: Faceted Search. To allow iterative and incremental filtering of large information networks down to a few items of interest, a common solution is to provide keyword search functionality and the means to apply filters based on a facet type, facet value, and facet count. For example, in faceted search of a movies dataset, one facet type might be “Genre”, its values might include “Drama,” “Comedy,” and “Sci-Fi,” and each of these values may be accompanied by the count of movies in the dataset matching that facet value. Early approaches to faceted search, such as Flamenco [YSLH03], presented clickable facets that could be progressively applied as filters to “drill down” into a target dataset. Similarly, FaThumb [KRR*06] was the first to introduce faceted search on mobile devices. Visualizations have also been introduced to the faceted search experience as an aid navigation. For instance, PaperLens [LCRB05] features multiple views to formulate a limited set of publication data-oriented questions, such as the relationships between authors or frequently referenced papers. FacetMap [SCM*06] is a domain-agnostic generalization of PaperLens that supports simple searching and browsing tasks using a tree-map that scales with display size. FacetLens [LSR*09] extends this line of work further by exposing facet values as visualizations such as bar charts whose elements can be selected directly.

Our approach with Orchard extends the functionality of these systems in several important ways. First, facet values are shown in a list-based view, allowing users to sort either alphabetically by facet value or numerically by facet count and other aggregations. Second, users can select multiple facet values from such lists and choose to match either any or all of these values. Third, by incorporating network projection into the analysis of heterogeneous networks, users can retrieve potentially interesting combinations of facet values more easily and systematically.

CC2: Topological Search. Researchers have devised various visual query languages to express network queries (e.g., [BCZ13, CLLT15, CFT*08]). A recent example of such a system is Visage [PHT*17], a visual graph querying approach in which users

iteratively refine attribute and topological constraints on a 2D layout. The system populates user-defined patterns with examples and allows them to browse other possible matches.

Unlike Visage, Orchard does not require users to define topological constraints by specifying a graph, but instead lets them explore aggregated nodes simply by following links.

CC3: Node and Link Aggregation. A common issue when dealing with node-link representations of multivariate networks is space and legibility – such representations can easily turn into undifferentiated hairballs for large, dense networks [ELMW18]. To visualize arbitrarily-sized networks compactly, nodes and edges can be aggregated rather than represented individually. Aggregation is typically applied on the attribute level, e.g., meta-nodes representing all nodes with a given categorical attribute value or numeric attribute range. For example, PivotGraph [Wat06] aggregates and positions nodes based on their attributes in a grid, encoding the relationship between node aggregates by the thickness of visual links. Similarly, OntoVis [SMER06] uses information in the ontology associated with a network to semantically prune a large, heterogeneous network. Juniper [NSL18] lets users explore the relationships of nodes in selected sub-networks using tree visualization, where aggregated nodes can be collapsed and expanded. Finally, both GraphTrails [DHRL*12] and work by Van den Elzen & Van Wijk [VdEVW14] use common data visualizations such as bar charts and tag clouds to summarize subsets of nodes.

Orchard's approach to aggregation is similar to GraphTrails' in that users can choose how facet values of a particular node type are grouped and aggregated. However, in addition to showing count of nodes per group, Orchard provides additional metrics, such as the count of nodes of a linked node type, that users can sort by.

CC4: Pivoting to Linked Nodes. Pivoting is complementary to searching and filtering as it expands, rather than reduces, the options for onward navigation. Given a subset of nodes, users can choose to “pivot” to view linked nodes, e.g., those of a specified type. Various styles of pivoting between node types have been proposed in the literature. FacetLens [LSR*09] provides a one-to-many pivot mechanism by displaying pivot options for a selected node. In PivotSlice [ZCCB13], users can configure a 3-by-4 multiple focus view that sub-divides and visualizes network data based on user-defined facets. PivotPaths [DRRD12] enables pivoting between partially overlapping sets of resources and facet values, and attempts to make pivots more comprehensible by using animated transition to highlight the overlap in elements between successive views. GraphTrails [DHRL*12] introduces a many-to-many pivot mechanism that allows users to pivot from a subset of nodes to the set of directly connected nodes, providing a freeform canvas on which users can lay out multiple pivot paths in 2D space.

Orchard adopts a similar kind of many-to-many pivoting as GraphTrails. However, in Orchard users can freely navigate from a set of nodes of a particular type to directly or indirectly linked nodes of *any* other node type in the network, preventing them from having to fully understand the underlying network structure and from getting stuck in leaf nodes.

CC5: Projections of Sub-networks. Given a sub-network, it is often insightful to analyze interactions between the nodes of a particular type as a homogeneous network. A commonly used technique is to “project” a heterogeneous network to a homogeneous one by connecting two nodes of the same type if both are connected to the same node of a different type [LMDV08]. Systems such as Orion [HP14] and Ploceus [LNS14] are built to support users in formulating such transformations as a pre-processing step: users can map tabular data to a network structure, by defining the semantics of links between nodes. Such projections are more readily interpretable as node-link representations than the full heterogeneous network. Typical analyses of homogeneous networks include the extraction of connected components and communities of nodes that are preferentially attached to one another.

Orchard does not make this assumption, but instead promotes on-demand projection of any node subset to homogeneous networks, all as part of an exploratory workflow rather than as a pre-processing step.

2.2. Data Exploration on Touch-Enabled Devices

Designing visual interfaces for exploring data on touch-enabled devices is becoming an increasingly important topic within the information visualization community [LIRC12, LBI*18, CDIL19, LCI*20]. For example, PanoramicData [ZZD14], Vizdom [CGZ*15], and SketchInsight [LSR*15] featured a pen and touch interface for visual data exploration on large interactive displays. Tangraphe [TSS18] proposed a set of single hand, multi-touch gestures for interactive exploration of network visualizations, while Schmidt et al. presented a set of multi-touch interactions for network visualizations, focusing on edge interactions [SNDC10]. Prior work has also focused on smaller tablet devices. For example, TouchWave presented a set of multi-touch gestures to interact with a stream graph on tablet devices [BLC12]. Kinetica employed physics-based affordances with multi-touch interaction for multivariate data exploration on a scatterplot [RK14]. TouchViz evaluated a gestural interface for choosing, filtering, and sorting data in familiar charts on tablet devices [DFS*13]. Sadana and Stasko [SS16] have also explored challenges in creating multiple coordinated views in data visualization systems for tablet computers. Finally, TouchPivot [JLLS17] proposed a pen-and-touch user interface that aids visual data exploration on tablet devices, targeting novice users.

Orchard builds on this trend towards touch-enabled data exploration, focusing on single-touch gestures on mobile phones for the simplest and more broadly applicable interaction style for casual everyday activities.

3. Design Goals

Our research focuses on the casual exploration of multivariate heterogeneous networks, specifically in the context of mobile phones. We set out by designing and implementing various prototypes to support individual low-level tasks applicable to heterogeneous networks (Table 1), and by combining them to support end-to-end exploratory workflows. To derive a set of design goals we presented and tested early designs for solving these tasks on mobile phones

ID	Low-Level Task
L1	Determine the attributes and values associated with nodes
L2	Find the nodes with specific attribute values
L3	Find a derived property of a set of nodes with specific attribute values
L4	Find the set of entities adjacent to or accessible from an entity
L5	Find a derived property of the entities adjacent to or accessible from an entity
L6	Find the entity with the maximum/minimum number of adjacent entities
L7	Common connection: Given a set of entities, find a set of entities that are connected to all of them
L8	Follow path
L9	Revisit an entity and infer further knowledge
L10	Characterize sets of nodes as belonging to different groups, based on node attributes

Table 1: A set of low-level tasks from Pretorius et al. [PPS14] that are applicable to heterogeneous networks.

with invited test users and colleagues at our research institution. We prepared various test datasets ranging from movie databases to social news feeds, academic publication databases, and publicly available communication networks. The challenges we faced while iterating on the designs with our test users can be broadly categorized into representational and functional challenges. On the one hand we sought to find suitable mobile-friendly representations for techniques that have been proposed in the literature in the context of network analysis for larger displays. On the other hand, we learned that in some cases users were interrupted in their flow due to missing functionality that has not been explored in prior research multivariate heterogeneous networks. Guided by these findings we identified two representational (DG1 and DG2) and two functional design goals (DG3 and DG4) that we address in this paper.

DG1: Introduce mobile-friendly list-based representation. Views and operations to explore networks should be exposed through a design suitable for the targeted form factor. With Orchard we aim to support casual exploration scenarios through a simple user interface design reminiscent of that of other information retrieval applications for mobile phones. Our goal is to introduce a new list-based paradigm with gestural navigation of large multivariate heterogeneous networks.

DG2: Support lightweight gestural navigation. Transitioning between different node types plays an important role in analyzing heterogeneous networks [DHRL*12]. However, understanding which nodes can be pivoted to and deciding on which nodes to pivot to places extra cognitive load on users. To promote free-form exploration, we aim to make pivoting a lightweight, single-touch operation that can easily be reverted, allowing users to quickly explore multiple pathways.

DG3: Consolidate drill-down filtering and lateral pivoting. During testing of early prototypes of Orchard we observed that users often select facets of interest of different node types while pivoting through the network, until at some point they want to apply multiple previously selected facets to the current set of nodes. Drill-down filtering in the style of faceted search can be used to quickly narrow down a collection of data items to a smaller subset, whereas lateral pivoting allows users to move between sets of nodes related through multiple hops. Our goal is to consolidate these complementary mechanisms by allowing users to turn a sequence of lateral pivot operations into facets that can be applied as filters, or vice versa.

DG4: Integrate network projections into exploratory analysis. Although heterogeneous and homogeneous networks can hardly be explored in the same view, we postulate that users benefit from on-demand network transformations as part of a regular network exploratory workflow. Being able to project arbitrary sub-networks of interest to homogeneous ones allows users to gain insights about such sub-networks from a different, customized perspective, in ways that can inform decisions on what to explore next in a heterogeneous context.

4. Pivot Trails

To retrieve the data necessary to render network nodes in a list-based visualization (DG1) users must be able to specify aggregation queries for subsets of nodes satisfying topological and attribute constraints. In this section, we introduce a query specification model called *pivot trails*, designed to capture common queries for nodes in multivariate heterogeneous networks, and capable of consolidating drill-down filtering and lateral pivoting (DG3).

4.1. Data Assumptions

Our pivot trails model relies on three assumptions on the data.

First, every node instance of a node type in a heterogeneous network can have an arbitrary number of attributes/facets.

Second, the network is multi-partite (k -partite), meaning that it can be divided into k independent sets, such that no link connects two nodes of the same type. Consider, for instance, Figure 1 showing how different entities in a film dataset and the relationships between them are represented in a k -partite network.

Third, links are homogeneous and univariate, meaning that there exists one link type only, with no associated attributes. Instead, a heterogeneous relationship is modeled by association to new node type. Consider, for instance, the relationship between `Person`, `Role`, and `Film` in Figure 1. An alternative schema could model `Role` as an attribute of a link connecting `Person` and `Film`. Yet, promoting heterogeneous relationships to their own node type obviates the need for the concept of *links* in a user interface, without the loss of any of the link semantics.

Furthermore, because the network is k -partite any set of nodes containing nodes of two different types can be projected to a homogeneous network. In Figure 1, for instance, the sub-network containing topics and films can be projected onto two different homogeneous networks: a `film` ↔ `film` network, where two films are

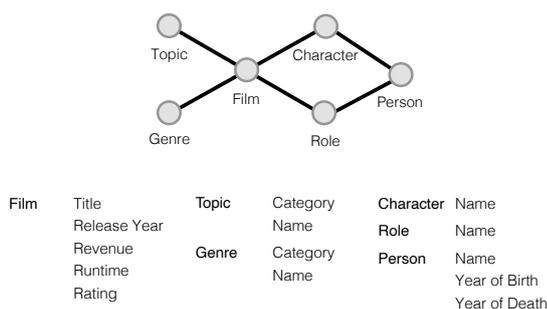


Figure 1: The schema of a multivariate heterogeneous network containing information about films, actors, and so on, as well as the attributes for every node type.

connected by a weighted link representing the count of common topics, or a $topic \leftrightarrow topic$ network, where two topics are connected by a weighted link representing the count of films about the two topics.

4.2. Query Specification Model

A typical aggregate network query consist of the following components [DHRL*12]:

- A set of *topological and node attribute constraints* that describe how to arrive at a subset of nodes of interest;
- A *grouping strategy* for the attribute of interest in the retrieved subset of nodes; and
- An *aggregation strategy* that defines which metric should be calculated for each bin.

With *pivot trails* we introduce a convenient way to model network queries as topological and node attribute constraints, applicable to a wide range of network exploration scenarios. Intuitively a pivot trail defines the order in which to traverse, filter, and combine nodes in a heterogeneous network to a set of nodes of interest, i.e., a query result. Pivot trails are directed graphs that can easily be translated to graph query languages such as Cypher, or even SQL.

Formally, assume a k -partite heterogeneous network H where each node is of a node type t_m where $m \in \{0, 1, \dots, k-1\}$ (Figure 2a). A pivot trail is a directed acyclic graph. Each node P_i in a pivot trail (*pivot node*) has a node type $T(P_i)$, a set of node-attribute constraints (filter predicates) $C(P_i)$, and a set operation $O(P_i)$. A pivot trail contains exactly one target node P_n , where $T(P_n)$ is the node type of interest.

A pivot trail defines how nodes in H of a particular type are retrieved. The pivot trail shown in Figure 2b, for instance, translates to the following query: retrieve all nodes in H of type $T(P_0)$. For all nodes that satisfy constraints $C(P_0)$ traverse their links leading to a set of nodes of type $T(P_1)$. For all nodes that satisfy constraints $C(P_1)$ traverse their links leading to a set of nodes of type $T(P_2)$, and so on, until the target pivot node is reached.

Pivot Trails are designed to express both lateral pivoting (navigating from a subset of nodes of type a to a set of linked nodes of

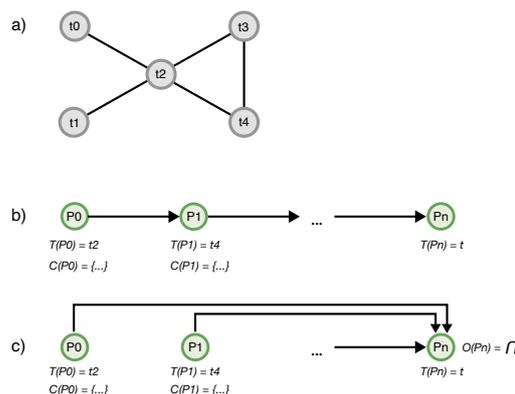


Figure 2: a) An example schema of a heterogeneous network with node types $t_0 - t_4$; b) a pivot trail representing a lateral pivoting operation; and c) a pivot trail where all pivot nodes are directly linked to the target pivot, representing a drill down operation (set operations for pivot nodes with one or no incoming links were omitted)

type b , where $a \neq b$) and drill down filter operations (navigating to a set of nodes that are linked to m other sets of nodes, where $m > 1$), or combinations thereof. Figure 2b, for instance, represents a sequence of lateral pivot operations. Conversely, this pivot trail can be re-structured to a drill down operation by linking any or all $n - 1$ pivot nodes directly to the final pivot node P_n and setting its set operation $O(P_n)$ to \cap , as shown in Figure 2c.

A pivot trail graph does not need to be fully specified, but can be inferred from a partial specification (at the minimum a target pivot node must be given). In contrast to traditional query models proposed in the literature, which only consider adjacent nodes as valid pivot paths, this allows users to formulate queries in a more flexible way. For instance, given the schema in Figure 1, people can be directly related to genres or topics, without the need to specify intermediate hops. Figure 3 shows a number of concrete examples of user-defined pivot trails and how full pivot trails are inferred from partial ones using the schema in Figure 1.

5. Orchard

Guided by the four design goals (DG1-4) presented in Section 3 we built a research prototype, called *Orchard*. To inform the design of a visual exploration user interface that incorporates the five core concepts (CC1-5) of network exploration outlined in Section 2.1, we carried out an iterative design process (Section 3). In this section, we present the final design of Orchard.

5.1. Implementation

Orchard is a mobile application written in TypeScript and HTML/CSS, built to run on all modern mobile phones. It uses a dynamic layout that fits the screen of the phone and is designed to be used in portrait mode. Orchard is backed by a C#-based implementation of the query model we propose in Section 4.

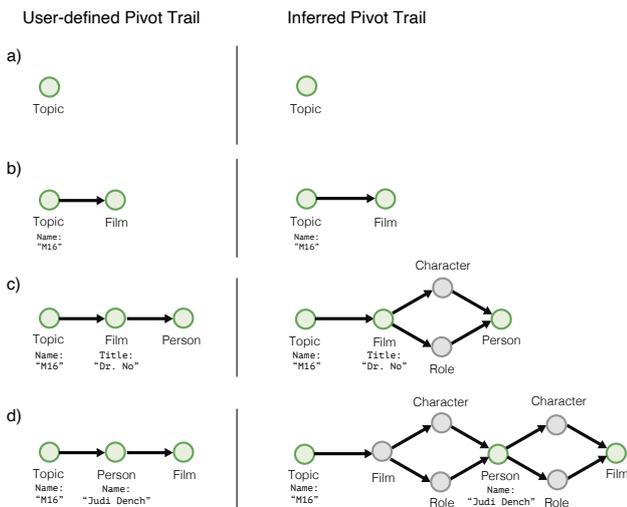


Figure 3: Examples of pivot trails. a) find all topics; b) find all films associated with a topic whose name includes ‘MI6’; c) of films related to a topic containing ‘MI6’ in its name, find all people linked to the films containing ‘Dr. No’ (note that the intermediate nodes required to reach *Person* from *Film* are not explicitly specified, but inferred from the network schema); and d) find all the films that are linked to person nodes containing ‘Judi Dench’ in their name, which are linked to a topic containing ‘MI6’ in their name.

5.2. Attribute Exploration

The *Attribute Exploration View* (Figure 4), Orchard’s default view, is to browse aggregates of nodes in a network. Every item in the list represents an aggregate (a category for a categorical attribute or a range for a numerical attribute) defined by the grouping and aggregation strategy (Figure 4a,b). List items contain a bar whose length encodes the aggregated value per bin, such as the count of all nodes linked to the nodes in a bin, the average of one of their numerical attributes, etc. Tapping on one of the list items reveals more details about the nodes in that group. By tapping the grouping strategy (Figure 4a), users can switch the node type of the current pivot node and pick a facet of interest. Similarly, the aggregation strategy can be changed by tapping on the second column (Figure 4b). There are two types of aggregate values that can be computed:

1. The count of distinct linked nodes of a certain type (e.g., the number of films per actor name, how many films have the same title or a rating between 90 and 100).
2. The min/mean/max of linked numeric node attributes (e.g., the mean revenue of all “Alice in Wonderland” films, the average age of all people per film).

A default grouping and aggregation strategy can be configured per attribute per node type. For instance, users can choose to always display the count of associated films when looking at genre, or the count of associated people when looking at characters. Furthermore, lists can be sorted by the facet value (Figure 4a) or by the aggregated value (Figure 4b).

5.3. Visual Query Interface

To support network exploration we designed a visual query interface that implements the pivot trail model. Orchard exposes a visual representation of a pivot trail as horizontal list at the top of the screen. On start up, a single pivot node of a configurable default node type is shown in the pivot trail, which corresponds to the data visualized in the *Attribute Exploration View*. The pivot bar on the right of the screen shows all pivot node options, and scrolls vertically if the number of pivot nodes exceeds the height of the screen. Users can add a pivot node to the end of the pivot trail (Figure 4d) by performing a “swipe left” gesture on the pivot node. As the finger slides across the display while swiping, the current list moves to the left and visual feedback indicates which type of pivot node users are about to add to the pivot trail (Figure 4).

In open-ended browsing sessions, we observed that users often select facets of interest of different node types while pivoting through the network, until at some point they want to apply multiple previously selected facets to the current set of nodes. Orchard supports this browsing behavior (*DG3*), allowing users to choose between lateral pivoting and drill down filtering by changing how nodes in the pivot trail are linked. By default, all nodes in a trail are linked sequentially, analogous to the depiction in Figure 2b. To switch from lateral pivoting to drill down filtering, users can link individual pivot nodes to the target pivot node by pressing the link icon to the right of each pivot node. An activated link indicates the corresponding pivot node is linked directly to the target pivot node, a de-activated link indicates that the pivot node is linked to its successor in the trail.

By pivoting from subsets of nodes to new subsets, users keep adding pivot nodes to the pivot trail. They can navigate back and forth in the pivot trail by swiping left or right, analogous to traversing an undo/redo stack. They can also jump to or remove a specific node, or start a new trail from an existing pivot node by using the context menu that appears when tapping a node in the pivot trail. Furthermore, users can constrain the subset of nodes from which to pivot by selecting list items, and select whether *all* or *any* of the selected items should be matched via the context menu. Pivoting from a list of nodes where no selection has been made is treated as if all items were selected.

5.4. Keyword Search

A common strategy to explore large graphs is a bottom-up approach, where the analysis begins with a search and more context is added as needed [VHP09, VLKS*11]. Orchard provides keyword search functionality where partial matches of node attributes are displayed in a list (Figure 5A). Users can navigate to the matched nodes by tapping on one of the results, which initializes the pivot trail with the required pivot node and constraints.

5.5. Reflective Pivots

A special type of pivot operation are *reflective pivots*, i.e., pivoting from/to the same node type. As there are no self-referential nodes in *k*-partite networks, this interaction is, in theory, meaningless. With Orchard we address this semantic gap by treating two consecutive

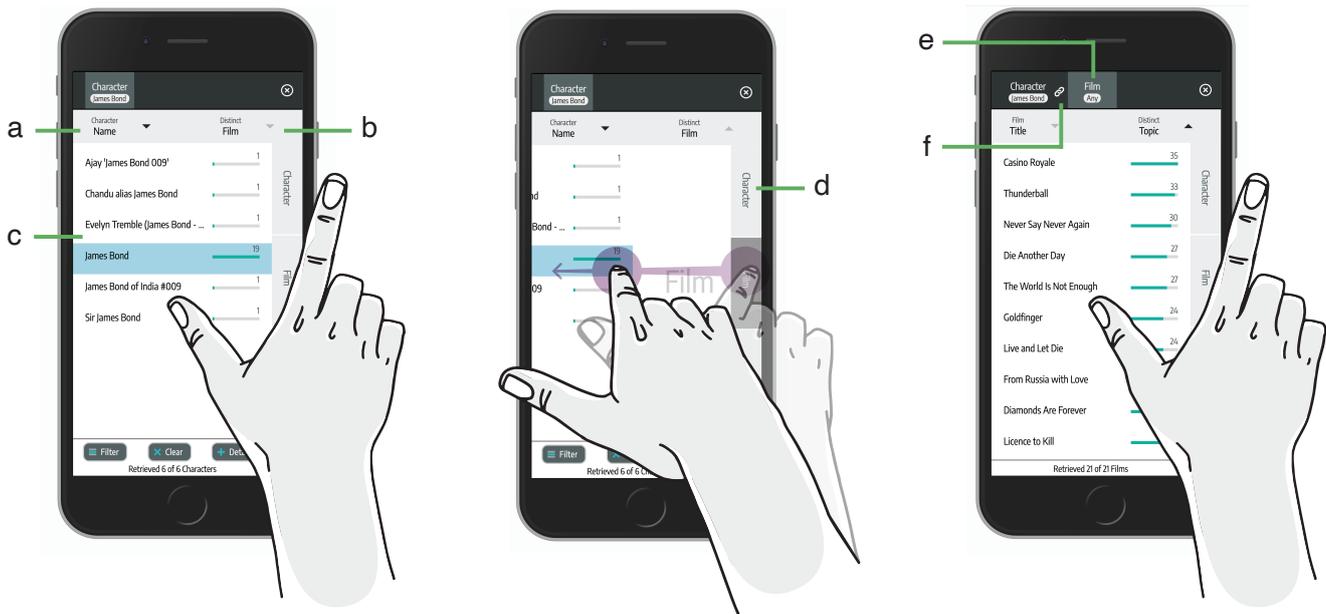


Figure 4: Orchard's facet exploration view and swipe-based pivot mechanism. Users can configure how nodes are grouped (a), aggregated (b), and then visualized in the bar chart-like list (c). The pivot bar (d) displays all node types of a dataset as tabs along the right edge of the application. Starting a swipe-left gesture on one of these node types moves the current view to the left and pivots to all nodes in the network that are linked to nodes in the selected group (films linked to characters named “James Bond”). Upon completion of the gesture, a new instance of the swiped node type is added to the pivot trail (e) and the result list is updated. Since the first pivot node is always directly connected to the target pivot node when there are only two nodes in the pivot trail, the link between them is activated (f).

pivot nodes with the same type as an instruction to project their matching nodes to a homogeneous network ($DG4$). Such a projection generates a link between any pair of matched nodes, if they are linked to the same node of another node type specified by users. The assigned link weight between such a pair of nodes is defined by the number of distinct nodes to which the pair is connected in their heterogeneous context. Since network projections are computationally expensive, Orchard computes the results in a progressive fashion, giving users the ability to refresh the results on demand. The result of a projection is displayed as a link list comprising pairs of facets values and counts indicating how frequently that combination occurs (the link's weight). The link semantics can be changed by changing the node type used in the aggregation strategy, allowing users to create arbitrary projections of interest. For example, as Figures 5D-F illustrate, a repeated swipe on `Character` leads to a projected network shown as an edge list in Figure 5E. Changing the aggregate pivot, for instance, to `Topic`, will re-project the current sub-network, creating pairs of `Character` nodes if they are connected to the same topic. The edge list behaves analogously to the node list; items are sorted based on the link weights, and pairs of node attributes can be selected as constraints, i.e., filter predicates.

6. Applicability

Orchard's design facilitates domain-agnostic, casual exploratory workflows of multivariate heterogeneous networks. To demonstrate its broad applicability, we tested Orchard on a number different datasets, ranging from politics, social media, to sports and cooking.

In this section, we briefly highlight three exploration examples and detail the steps of a fictional usage scenario using a movie dataset (also depicted in the supplementary video).

6.1. Usage Scenarios

Given a dataset containing political lobbying information of the city of Chicago [chi19], our users found answers to questions such as: who are prolific or unsuccessful congress members? Who do they work with? What are common topics in the bills they endorse? Who endorses similar bills? Is there bias in different committees?

Using “likes”-data from Facebook [RDSS18], we set out to enable users to explore how different artists and public figures relate to companies, food, sport and TV shows. We annotated the data with additional information through public knowledge graph APIs. For example, we added descriptions such as *singer/songwriter* for Taylor Swift or *Fast-Food company* for McDonald's. Browsing this dataset, users were able to quickly identify unexpected and entertaining commonalities between celebrities.

Finally, from a publicly available soccer dataset [Mat16] we extracted information about entities such as players, matches, leagues, and teams. Our users were interested in answering questions such as: which players have played against each other most frequently? How many teams have they played for? Have these players ever played on the same team? Which leagues transfer most players among each other?

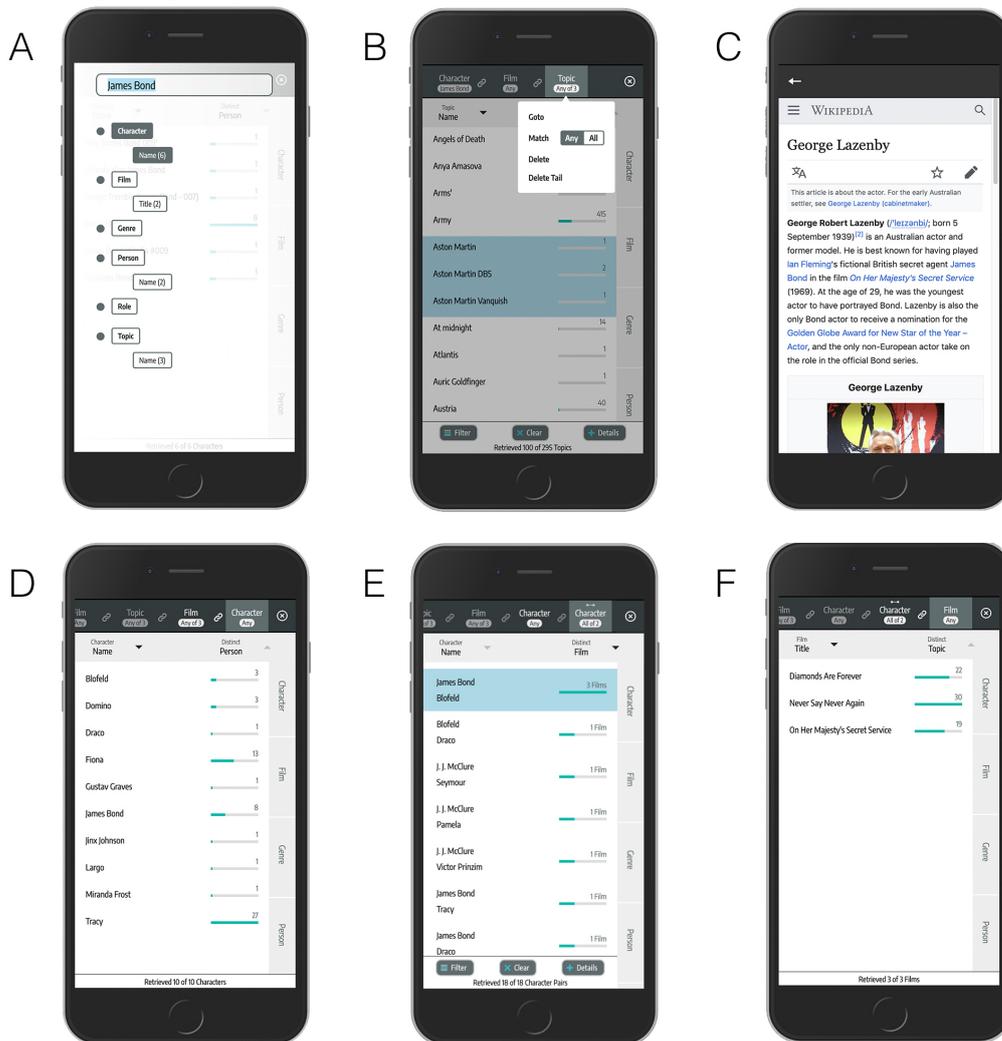


Figure 5: Different views in Orchard. (A) Keyword search with result listing; (B) The context menu of a pivot node in the pivot trail; (C) An example of customizable detail view for a specific node, in this case the Wikipedia page of an actor; (D,E) Shows an example of a reflective pivot: swiping from Character to Character projects a subset of network nodes shown in (D) to a homogeneous Character \leftrightarrow Character network (E). The result of this operation is a list showing a selected attribute (character name) of pairs of nodes (E). Each pair shares one or more linked nodes of a selected type (Film in this case). Selecting a pair of nodes in this view is equivalent to selecting both nodes individually such that they can be used as facets for subsequent views (F).

6.2. Film Dataset Use Case

Emily - a film enthusiast - heard rumors about a new James Bond film being released soon. Knowing that Daniel Craig will be starring in the new Bond film, she is curious to learn more about other Bond films, associated actors and actresses, their roles, and so on. On her commute back from work Emily launches Orchard on her mobile phone and selects a dataset containing information about films and TV shows, such as title, release year, ratings, associated people, characters, and topics (see Figure 3 for more details).

What film characters are named “James Bond”? Emily first uses keyword search for “James Bond” and sees multiple potential facet types in which to search for a match (Figure 5A). She decides

to match against characters named “James Bond,” and then find all associated films. To do so, she selects the Character node type from the search results (Figure 5A) and picks the exact match, which also has the greatest number of associated Film node types (Figure 4). To pivot to all films associated with the James Bond character, she touches the Film tab on the right and swipes left to reveal the results.

Which James Bond film has the highest rating? Emily changes the aggregation strategy, setting it to Film.Rating. She then sorts the results in descending order by tapping the sort icon next to the aggregate arrow next to the aggregation strategy.

What topics are associated with James Bond films? Curious about the topics associated with any of these films, she swipes left from the `Topic` tab. While scrolling through the list of James Bond film topics, she finds that there are three different kinds of “Aston Martin” topics, with 1, 2, and 1 linked films, respectively (Figure 5B). But how many of them are James Bond films? She swipes left from the `Film` tab to see all films associated with these topics. She then taps the link icon next to the prior “James Bond” `Character` selection to add this facet to the current filter. As a result, all non-Bond films are removed from the results list.

Which actors played James Bond in these films? Emily swipes left again from `Person` to get a list of actors. She knows Pierce Brosnan, Roger Moore, and Sean Connery, but who is George Lazenby? To find out she taps on the “Details” button at the bottom, which takes her to a detail view, in this case George Lazenby’s Wikipedia page (Figure 5C).

Which Bond characters frequently appear together? Emily goes back to the previous view by swiping right, and then swipes left from `Character` to see all characters associated with Bond films (Figure 5D). To learn which of these characters frequently co-occur in the same film, she swipes left from `Character` again to project the current heterogeneous sub-network (`Character` ↔ `Film`) to a homogeneous network (`Character` ↔ `Character`). She finds that James Bond and Blofeld are the most frequently occurring pair, co-occurring in three different Bond films (Figure 5F).

7. Evaluation

To assess the utility of Orchard, to learn how quickly users are able to grasp how to formulate common network queries, and to uncover potential usability issues in Orchard, we conducted a lab study with 12 participants (4 females). All participants were graduate students and all except one reported to be at least “slightly interested” in films and TV shows.

7.1. Dataset and Tasks

We designed our study based on similar experiments carried out in prior work [ZCCB13]. As most people have at least some familiarity with films or TV shows, we decided to create a custom movie dataset containing data from IMDB (www.imdb.com/interfaces) joined with with topics extracted using our own algorithms from film plots in the CMU Movie Summary Corpus [BOS13]. We then developed a series of tasks for evaluating Orchard. The tasks are inspired by Pretorius et al.’s task taxonomy for multivariate network analysis [PPS14] which is based on work by Lee et al. [LPP*06] and Amar et al. [APS13]. We considered only those tasks that are equally applicable to homogeneous and heterogeneous networks alike (Table 2).

7.2. Procedure

We began with a five-minute demonstration of the features of Orchard by walking the participants through a scenario similar to the introductory use case described in Section 6.2. The demonstration included knowledge required to accomplish the eight tasks we created for this evaluation. Following the introduction we handed the

ID	User Study Task	Low-Level Tasks
T1	In which year was the film “The Shawshank Redemption” released?”	L1, L2
T2	Which people were involved with this film?	L3, L7, L8
T3	Of the people involved with this film, who contributed most to other films?	L4, L5, L6
T4	Of the people involved with this film, who contributed to the most distinct genres?	L4, L5, L6
T5	How many writers contributed to the film?	L6, L8
T6	Who were these writers?	L3, L4, L5, L8
T7	Which other films have these writers worked on together?	L3, L4, L5
T8	Which two people associated with this film are the strongest collaborators?	L9, L10

Table 2: Eight tasks we asked used in our user study, and their corresponding low-level tasks (shown in Table 1).

mobile phone over to the participants and asked them to repeat the same or a similar sequence of steps. The participants were encouraged to ask questions during this initial phase. In the second part of the study we instructed participants to complete eight tasks we prepared (Table 2). They were given two minutes to solve each task without any help. If they were not able to find an answer in two minutes, we showed them how to get to the solution. In the third part of the study we asked participants to freely explore the dataset. During this process, participants were encouraged to think-aloud, i.e., to narrate their actions, intentions, and reactions. Finally, we gave participants a questionnaire and conducted semi-structured interviews to collect their feedback. Each session lasted approximately 50 minutes.

7.3. Results

In the second part of the study the majority of our participants were able to complete all eight tasks quickly (Figure 6), given that they had only gone through approximately ten minutes of introduction and training that covered a total of 14 operations to prepare them for the tasks. More than a half (56.3%) of all tasks were completed in less than 30 seconds, and 21.8% in less than 10 seconds. In only 14.6% of all cases, participants exceeded the two-minute limit and were offered assistance. In T1, for instance, some participants struggled to find the keyword search or the button to get to the details of the selected film(s). In T3, some participants forgot how to change the aggregation strategy to configure what is being shown in the bar chart. However, once they found out they were able to complete a similar task much faster (T4). A task of similar complexity was T5, which required participants to use and configure three pivot nodes, as well as the aggregation strategy. Out of all twelve participants, only one was unable to correctly apply facets such that the count of writers was restricted in the manner directed. Similarly, only one participant needed assistance in T6 because he did not remember he could swipe left on person to see the people’s names associated with a selected role. In T7, some participants forgot how

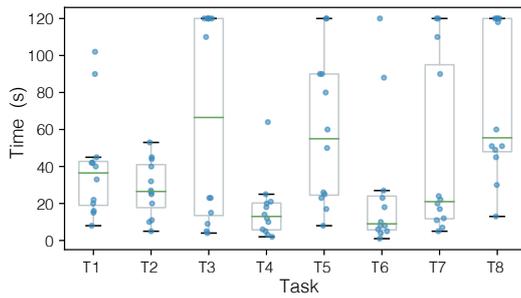


Figure 6: Task completion time for each task.

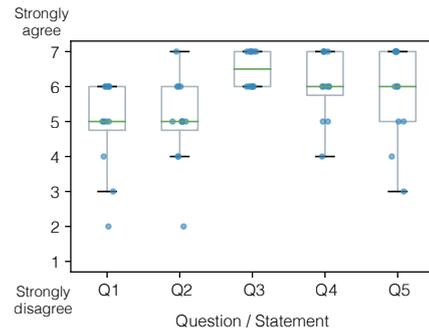
to change the boolean operator from matching any selections, to match all selections. Finally, most participants did not immediately recall how to create pair-wise combinations, but were able to figure it out in less than a minute.

In the third part of the study we asked participants to freely explore the dataset. We made the following key observations:

- *Intuitive swiping*: Participants were immediately comfortable navigating back/forth by swiping left or right. Swiping on a related entity to pivot was reported to feel natural and intuitive.
- *Long pivot trails*: The number of nodes in a pivot trail during exploration can grow rapidly. Various participants pointed out that it would be helpful to be able to store intermediate results by collapsing a (sub-)trail into a named pivot node.
- *Difficulty in distinguishing between lateral pivoting and drill-down filtering*: Although most participants were able to understand the semantics of Orchard quickly through trial-and-error, we observed that it was not immediately obvious to participants which facets were applied to the current view, i.e., how activating links in the trail affect the result in the current view.
- *Opportunity in exposing the computational path on demand*: While our query model hides the complexity of queries that involve multiple hops, some participants mentioned that exposing the computational path on demand would help users gain a better understanding of the network's topology.
- *Configurable defaults*: In some cases users would have preferred different defaults (e.g., the default aggregation strategy per node type, the order of node types in the swiper).

When completing a questionnaire of five questions using a 7-point Likert scale (1: “Strongly disagree” to 7: “Strongly agree”; Figure 7), despite the minimal training, most participants indicated that Orchard was easy to learn and use (Q1 and Q2). Participants also indicated that Orchard helped them find new and re-discover previously known insights (Q3 and Q4), and that swiping aided orientation while navigating through the network (Q5).

In addition, we overall received encouraging feedback: “Once I got used to the defaults it started to become very intuitive.”; “I liked the trails because it was very visual, you can think about everything as a line, everything is like a chain.”; “Swiping to navigate felt very intuitive.”; “Exploration was easy, you get where you want to.”; and “I see this broadly applied to other datasets.”



- Q1: The system was easy to learn
 Q2: The system was easy to use
 Q3: The system helped me find insights I didn't know about before
 Q4: The system helped me confirm/re-discover things I knew before
 Q5: Swiping left/right was helpful for orientation

Figure 7: Results of our post-study questionnaire.

8. Future Work

While we have received encouraging feedback, there are a number of intriguing opportunities for extensions and improvements. Orchard currently supports on-demand network projections for any pair of node types, for any sub-network. Although our prototype shows weighted links for any combination of two nodes aggregates, we believe that users would benefit from a tighter integration with homogeneous projections. More specifically, Orchard could extract entire groups of nodes from projected (sub-)networks, i.e., by computing clusters/communities, connected components, etc. Homogeneous projections also open up new possibilities to add meta-data to nodes that could be used for actions like sorting. For instance, users could sort a list of nodes by common metrics used in network analysis, such as degree- or betweenness centrality, rather than just an aggregated facet value or facet count. However, blending-in concepts from homogeneous networks in workflows for exploring heterogeneous networks also poses challenges: how can we better communicate what projected networks represent, and how can we convey what complex networks metrics indicate in a way that non-expert users can understand? These questions would be useful to pursue in future design iterations.

9. Conclusion

We have presented Orchard, a mobile application that uses a novel query model to facilitate a simple user interface design to explore multivariate heterogeneous networks. Using our approach, minimal user input is required to formulate common network queries. With Orchard, users can fluidly navigate through multivariate heterogeneous networks, using gesture-based mechanisms to create facet trails, to pivot to entities of interest, and to collect and apply facets as part of an exploratory workflow. Given the ubiquity of information networks and the encouraging feedback we received from experts and lay users, we see great potential in providing a mobile solution for casual browsing of richly associated datasets.

References

- [APS13] AHN J.-W., PLAISANT C., SHNEIDERMAN B.: A task taxonomy for network evolution analysis. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 365–376. 2, 9
- [BCD*10] BEZERIANOS A., CHEVALIER F., DRAGICEVIC P., ELMQVIST N., FEKETE J.-D.: Graphdice: A system for exploring multivariate social networks. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 863–872. 1
- [BCZ13] BHOWMICK S. S., CHOI B., ZHOU S.: Vogue: Towards a visual interaction-aware graph query processing framework. In *CIDR* (2013). 2
- [BLC12] BAUR D., LEE B., CARPENDALE S.: Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of ACM ITS '12* (2012), pp. 255–264. 3
- [BOS13] BAMMAN D., O'CONNOR B., SMITH N. A.: Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2013), pp. 352–361. 9
- [CDIL19] CHOE E. K., DACHSELT R., ISENBERG P., LEE B.: Mobile data visualization (dagstuhl seminar 19292). *Dagstuhl Reports* (2019). 3
- [CFT*08] CHAU D. H., FALOUTSOS C., TONG H., HONG J. I., GALLAGHER B., ELIASSI-RAD T.: Graphite: A visual query system for large graphs. In *2008 IEEE International Conference on Data Mining Workshops* (2008), IEEE, pp. 963–966. 2
- [CGZ*15] CROTTY A., GALAKATOS A., ZGRAGGEN E., BINNIG C., KRASKA T.: Vizdom: interactive analytics through pen and touch. *Proceedings of the VLDB Endowment* 8, 12 (2015), 2024–2027. 3
- [chi19] Chicagolobbyists.org. <http://chicagolobbyists.org/about>, 2019. Accessed: 2019-09-06. 7
- [CKHF11] CHAU D. H., KITTUR A., HONG J. I., FALOUTSOS C.: Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2011), ACM, pp. 167–176. 1
- [CLLT15] CAO N., LIN Y.-R., LI L., TONG H.: g-miner: Interactive visual group mining on multivariate graphs. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 279–288. 2
- [DFS*13] DRUCKER S. M., FISHER D., SADANA R., HERRON J., ET AL.: Touchviz: a case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), ACM, pp. 2301–2310. 3
- [DHRL*12] DUNNE C., HENRY RICHE N., LEE B., METOYER R., ROBERTSON G.: Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2012), ACM, pp. 1663–1672. 1, 3, 4, 5
- [DRRD12] DÖRK M., RICHE N. H., RAMOS G., DUMAIS S.: Pivot-paths: Strolling through faceted information spaces. *IEEE transactions on visualization and computer graphics* 18, 12 (2012), 2709–2718. 3
- [ELMW18] EDGE D., LARSON J., MOBIUS M., WHITE C.: Trimming the hairball: Edge cutting strategies for making dense graphs usable. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 3951–3958. 3
- [ELW18] EDGE D., LARSON J., WHITE C.: Bringing ai to bi: enabling visual analytics of unstructured data in a modern business intelligence platform. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), ACM, p. CS02. 1
- [HP14] HEER J., PERER A.: Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. *Information Visualization* 13, 2 (2014), 111–133. 3
- [JLLS17] JO J., L'YI S., LEE B., SEO J.: Touchpivot: blending wimp & post-wimp interfaces for data exploration on tablet devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), ACM, pp. 2660–2671. 3
- [KPLB07] KANG H., PLAISANT C., LEE B., BEDERSON B. B.: Netlens: iterative exploration of content-actor network data. *Information Visualization* 6, 1 (2007), 18–31. 1
- [KRR*06] KARLSON A. K., ROBERTSON G. G., ROBBINS D. C., CZERWINSKI M. P., SMITH G. R.: Fathumb: a facet-based interface for mobile search. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 711–720. 2
- [LBI*18] LEE B., BREHMER M., ISENBERG P., CHOE E. K., LANGNER R., DACHSELT R.: Data visualization on mobile devices. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–8. 3
- [LCI*20] LEE B., CHOE E. K., ISENBERG P., MARRIOTT K., STASKO J.: Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications* 40, 2 (2020), 82–90. 3
- [LCRB05] LEE B., CZERWINSKI M., ROBERTSON G., BEDERSON B. B.: Understanding research trends in conferences on paperlens. In *CHI'05 extended abstracts on Human factors in computing systems* (2005), ACM, pp. 1969–1972. 1, 2
- [Lea19] LEAKS I. O.: Panama papers. <http://offshoreleaks.icij.org>, 2019. Accessed: 2019-09-06. 1
- [LIRC12] LEE B., ISENBERG P., RICHE N. H., CARPENDALE S.: Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2689–2698. 3
- [LJH13] LIU Z., JIANG B., HEER J.: immens: Real-time visual querying of big data. *Computer Graphics Forum (Proc. EuroVis)* 32 (2013). URL: <http://vis.stanford.edu/papers/immens>. 1
- [LMDV08] LATAPY M., MAGNIEN C., DEL VECCHIO N.: Basic notions for the analysis of large two-mode networks. *Social networks* 30, 1 (2008), 31–48. 3
- [LNS14] LIU Z., NAVATHE S. B., STASKO J. T.: Plocus: Modeling, visualizing, and analyzing tabular data as networks. *Information Visualization* 13, 1 (2014), 59–89. 3
- [LPP*06] LEE B., PLAISANT C., PARR C. S., FEKETE J.-D., HENRY N.: Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on Beyond time and errors: novel evaluation methods for information visualization* (2006), ACM, pp. 1–5. 2, 9
- [LSR*09] LEE B., SMITH G., ROBERTSON G. G., CZERWINSKI M., TAN D. S.: Facetlens: exposing trends and relationships to support sensemaking within faceted datasets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), ACM, pp. 1293–1302. 1, 2, 3
- [LSR*15] LEE B., SMITH G., RICHE N. H., KARLSON A., CARPENDALE S.: Sketchinsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *2015 IEEE Pacific Visualization Symposium (PacificVis)* (2015), IEEE, pp. 199–206. 3
- [Mar06] MARCHIONINI G.: Exploratory search: from finding to understanding. *Communications of the ACM* 49, 4 (2006), 41–46. 2
- [Mat16] MATHIEN H.: European soccer database, Oct 2016. URL: <https://www.kaggle.com/hugomathien/soccer>. 7
- [Mee17] MEEKER M.: Internet trends report. https://www.kleinerperkins.com/files/INTERNET_TRENDS_REPORT_2017.pdf, 2017. Accessed: 2020-03-21. 1
- [NSL18] NOBRE C., STREIT M., LEX A.: Juniper: A tree+ table approach to multivariate graph visualization. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 544–554. 3
- [ope19] Opensecrets.org. <https://www.opensecrets.org/>, 2019. Accessed: 2019-09-06. 1
- [PHT*17] PIENTA R., HOHMAN F., TAMERSOY A., ENDERT A., NAVATHE S., TONG H., CHAU D. H.: Visual graph query construction and refinement. In *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), ACM, pp. 1587–1590. 2

- [PPS14] PRETORIUS J., PURCHASE H. C., STASKO J. T.: Tasks for multivariate network analysis. In *Multivariate Network Visualization*. Springer, 2014, pp. 77–95. 2, 4, 9
- [RDSS18] ROZEMBERCZKI B., DAVIES R., SARKAR R., SUTTON C.: Gemsec: Graph embedding with self clustering, 2018. [arXiv:arXiv:1802.03997](https://arxiv.org/abs/1802.03997). 7
- [RK14] RZESZOTARSKI J. M., KITTUR A.: Kinetica: naturalistic multi-touch data visualization. In *Proceedings of ACM CHI '14* (2014), pp. 897–906. 3
- [SA04] SHETTY J., ADIBI J.: The enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California* 4, 1 (2004), 120–128. 1
- [SCM*06] SMITH G., CZERWINSKI M., MEYERS B., ROBBINS D., ROBERTSON G., TAN D. S.: Facetmap: A scalable search and browse visualization. *IEEE Transactions on visualization and computer graphics* 12, 5 (2006), 797–804. 2
- [SH12] SUN Y., HAN J.: Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159. 2
- [SMER06] SHEN Z., MA K.-L., ELIASSI-RAD T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE transactions on visualization and computer graphics* 12, 6 (2006), 1427–1439. 3
- [SNDC10] SCHMIDT S., NACENTA M. A., DACHSELT R., CARPENDALE S.: A set of multi-touch graph interaction techniques. In *ACM International Conference on Interactive Tabletops and Surfaces* (2010), ACM, pp. 113–116. 3
- [SS16] SADANA R., STASKO J.: Designing multiple coordinated visualizations for tablets. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 261–270. 3
- [TSS18] THOMPSON J., SRINIVASAN A., STASKO J.: Tangraphe: interactive exploration of network visualizations using single hand, multi-touch gestures. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (2018), ACM, p. 43. 3
- [VdEVW14] VAN DEN ELZEN S., VAN WIJK J. J.: Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2310–2319. 1, 3
- [VHP09] VAN HAM F., PERER A.: “search, show context, expand on demand”: supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 953–960. 6
- [VLKS*11] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum* (2011), vol. 30, Wiley Online Library, pp. 1719–1749. 6
- [Wat06] WATTENBERG M.: Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 811–819. 3
- [WR09] WHITE R. W., ROTH R. A.: Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services* 1, 1 (2009), 1–98. 2
- [YSLH03] YEE K.-P., SWEARINGEN K., LI K., HEARST M.: Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2003), ACM, pp. 401–408. 2
- [ZCCB13] ZHAO J., COLLINS C., CHEVALIER F., BALAKRISHNAN R.: Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2080–2089. 1, 3, 9
- [ZZD14] ZGRAGGEN E., ZELEZNIK R., DRUCKER S. M.: Panoramic-data: Data analysis through pen & touch. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2112–2121. 1, 3