Proceedings Paper:
Pretorius, AJ, Purchase, HC and Stasko, JT (2014) Tasks for multivariate network analysis. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Dagstuhl Seminar , 12-17 May 2013, Dagstuhl Castle, Germany. Springer Verlag, 77 - 95. ISBN: 9783319067926. ISSN: 0302-9743. EISSN: 1611-3349.

https://doi.org/10.1007/978-3-319-06793-3_5

Tasks for multivariate network analysis

A. Johannes Pretorius[1], Helen C. Purchase[2], and John T. Stasko[3]

[1] University of Leeds, United Kingdom, a.j.pretorius@leeds.ac.uk
[2] University of Glasgow, United Kingdom, helen.purchase@glasgow.ac.uk
[3] Georgia Institute of Technology, United States, stasko@cc.gatech.edu

Abstract. In this chapter, we describe tasks that are typically encountered during visual multivariate network analysis. First, we present an overview of the entities and properties of multivariate networks and discuss a taxonomy for general visualisation tasks. We next describe a framework for multivariate network tasks and show how these tasks can be composed of lower-level tasks of the general taxonomy. We also include several real-world examples of multivariate network tasks as illustrations.

1 Introduction

In Chapter **??**, a multivariate network was defined as having two important characteristics. First, nodes are connected to each other via links; there is topological structure. Second, being multivariate, nodes and links have attributes associated with them, with these attributes having a value.

In this chapter, we describe tasks associated with multivariate networks. We consider a task to be an activity that a user wishes to accomplish by interacting with a visual representation of a multivariate network. This implies that there is user intent [1], and that the network has been presented visually. At the highest level, this intent is usually described as the goal of obtaining *insight* about the data being studied [2].

Pragmatically, the notion of gaining insight from visualisations can be described as one or more very high-level tasks. As Amar and Stasko put it, tasks that "real people want to accomplish" [3]. These include:

- Make complex decisions, especially under uncertainty;
- Learn a domain;
- Identify the nature of trends;
- Predict the future;
- Identify the domain parameters;
- Discover correlative models;
- Formulate and verify hypotheses;
- Identify the effect of data uncertainties; and
- Identify sources of causation.

In the spirit of Amar and Stasko's work, we note that this is a sample of high-level tasks and not a complete list. These tasks are not specific to multivariate networks and are biased towards exploration and confirmation. We recognise that some users may have additional objectives, such as the presentation of data, which fall outside the scope of this chapter. However, in a context where achieving insight is the main driver and where multivariate networks are of interest, performing a task such as those listed above involves one or more of the following activities [4]: gain an understanding of the structural properties of the network; find patterns, clusters, and correlations between several attributes of the nodes and links; and relate understanding about attributes and structure.

In this chapter, we describe in more detail how this is accomplished by presenting a framework of tasks for multivariate networks. Our objective is to present, to a general audience, a frame-of-reference that encapsulates the types of tasks typically encountered when analysing multivariate network data. As a result, the work presented here is deliberately not overly theoretical or abstract. We first recap the entities and properties of multivariate networks. We then describe a general taxonomy for visualisation tasks. Next, we introduce a framework for multivariate network tasks and show how these are composed of lower-level tasks of the general taxonomy. We follow with a short discussion before concluding.

2 Entities and properties

In the abstract, a task involves performing an analytic activity on a combination of an *entity* (the "thing" that is being studied), and a *property* of that entity [5]. We note that different terminology is sometimes used; for example, some authors refer to entities as data cases, and to properties as attributes [6]. Notwithstanding, a task can be represented as a process [5]:

Select entity → Select property → Perform analytic activity.

There is typically a high degree of iteration; based on the outcome of the analytic activity, the user may wish to select another entity and/or property to analyse. When considering multivariate networks, the entities that users study are [7]:

- *Nodes;*
- *Links;*
- *Paths,* or sequences of nodes and links; and
- *Networks,* since users may want to include several networks in their analysis.

Multivariate networks also have two types of associated properties [5]:

- *Structural properties,* sometimes referred to as topology; and
- *Attributes,* associated with nodes and links.

To make the above more concrete, we briefly revisit examples of multivariate networks from the three applications areas discussed in Chapters **??**, **??**, and **??**. In software engineering, analysts study entities including software packages, classes, and methods. Tasks include studying the links, such as method calls, between entities. Properties of nodes and links model features that are fundamental to understanding software including package, class, and method names, and method call durations. Multivariate networks in biomedicine include metabolic networks (nodes represent atoms, links represent bonds), interaction networks (nodes represent metabolites, links represent interactions), and regulatory networks (nodes represent proteins, links represent actions). Again, properties are important to facilitate insight, for example, whether actions in regulatory networks activate or repress protein production.

Social networks are perhaps more familiar to many readers (see **??**). In such a network, nodes represent people and links represent the relationships between people. By analysing paths between nodes, it is possible to derive knowledge. For example, even if two people have no direct relationship, if they both have a relationship with a third person, there exists an indirect relationship between them that may (or may not) be of interest. For social network analysis, there are scenarios where it is useful to compare networks themselves. For example, behavioural biologists may be interested in comparing social networks of humans with those of other primates to identify similarities and differences.

A lot can be learned from studying the properties of social networks. For example, it is possible to derive which of two people is likely to have the greater influence on others by considering, respectively, the number of relationships they have with other people. Properties provide important information, such as the type of relationships (friendship versus professional, for example) and demographics (first name, last name, occupation, and so forth).

Combinations of network entities and properties give rise to more complex concepts. For example, basic networks with structural properties only are less complex than networks with single node and link attributes (often referred to as labels) which, in turn, are less complex than multivariate networks where nodes and links can have multiple attributes. Increased complexity of networks results in increasingly complex analyses [8], and this impacts the complexity of tasks that are performed. In cases where users want to compare two or more networks, there is an additional level of complexity.

It is also possible to calculate derived entities and properties, that is, entities and properties that do not explicitly exist in the underlying data. Two common derived entities are clusters and groups. *Clusters* are regions of networks that are structurally highly connected (these are sometimes referred to as cliques, particularly in social network analysis). *Groups* are subsets of nodes and links that share similar attribute properties. Examples of derived properties include statistical measures computed for a particular attribute (mean, median, and so forth).

Derived entities and properties are often involved in multivariate network tasks. As suggested, in social network analysis, clusters indicate cliques, or collections of people who have a high degree of interaction. Grouping could be used, for example, to identify and compare sets of people with similar demographics.

3 Tasks

As highlighted above, tasks involve entities (nodes, links, paths, networks) and properties of those entities (structural and attributes). The third component that makes up a task is the analytic activity, or the analysis. Below, we deconstruct tasks by focusing on different levels of analytic activity. Throughout this, we also refer to the entities and properties that are involved in tasks in a multivariate network context.

We first outline a general taxonomy for interactive visualisation and then describe how some of these tasks are combined to form more complex tasks specific to multivariate networks.

3.1 General task taxonomy

Many authors have proposed general task taxonomies for information visualisation. In seminal work, Wehrend and Lewis propose a classification of visualisation methods by considering the entities being studied and tasks performed on the entities [9]. Specifically, they list 11 tasks that are frequently encountered:

- Identify;
- Locate;
- Distinguish;
- Categorise;
- Cluster;
- (Analyse) distribution;
- Rank;
- Compare;
- (Analyse) within and between relations;
- Associate; and
- Correlate.

By synthesising questions that users typically have about their data, Amar et al. propose a different list of information visualisation tasks [6]:

- Retrieve value;
- Filter;
- Compute derived value;
- Find extremum;
- Sort;
- Determine range;
- Characterise distribution;
- Find anomalies;
- Cluster; and
- Correlate.

In other related work on general information visualisation taxonomies, Schulz et al. recently proposed a classification of the "design space" of visualisation tasks based on five dimensions (*goal; means; characteristics,* or level of analysis; *target,* the parts of the data to be considered; and *cardinality,* the number of data instances to be considered) [10]. This allows a formal faceted specification of tasks by five-dimensional tuples. Brehmer and Munzner propose descriptions of visualisation tasks by considering three aspects [11]: *Why* is the data being analysed?; *How* is it being analysed?; and *What* are the task inputs and outputs? In particular, they stress the difference between how (the means) and why (the goal) a task is performed. There are clear parallels between these approaches: why relates to goal; how relates to means; and what encapsulates characteristics, target, and cardinality.

These approaches are very general and abstract (Shulz et al. write that theirs is "applicable by a limited number of visualization experts only") and do not easily support the definition of a detailed taxonomy; we return to them in the conclusion. More pragmatically, it is worth noting the similarities between the sets of tasks proposed by Wehrend and Lewis [9] and Amar and Stasko [3], for example, both make provision for studying distributions. However, a like-for-like comparison is not immediately obvious. Further, it could also be argued that both Wehrend and Lewis's and Amar and Stasko's list of tasks operate at varying levels, for example, a task such as "filter" is more of an operational task while "correlate" is more of an analytical one.

The work by Valiati et al. addresses such difficulties by distinguishing three broad classes of tasks [12]: operational (relating to the means by which the network is presented and explored), analytical (the means by which information is extracted from the network), and cognitive (facilitating understanding of the whole network). Each category comprises one or more tasks. Accordingly, the taxonomy put forward in this chapter is based on the categories and constituent tasks defined by Valiati et al., classified as follows:

– *Operational:* visualise, configure;
– *Analytical:* identify, determine, relocate, compare; and
– *Cognitive:* infer.

While we acknowledge the fundamental facilitating role that operational tasks play in making the relevant information visible, most of our emphasis will be on the analytical category. Cognitive tasks are also considered, keeping in mind that the purpose of the whole exercise is, of course, to support the cognitive task of obtaining insight (as described by the high-level tasks of Amar and Stasko [3] and listed in the introduction). To avoid additional complexity, we reuse the terminology proposed by Valiati et al., which in turn, is based on the work by Wehrend and Lewis, although some terms could arguably be substituted with other descriptive verbs.

Operational tasks. Operational tasks are concerned with the means of presenting the network to the user, and the facilities provided for the user to explore the data. These tasks are therefore more associated with the nature of the information visualisation tool than with the user's tasks per se.

– *Visualise.* Invoke a particular graphical representation or a combination of graphical representations to visualise the entities and properties of a multivariate network. The visualise task does not necessarily imply that all entities and all properties in the data are shown. In fact, it is almost always performed in combination with the configure task (described below) to selectively show or hide certain entities and/or properties.
– *Configure.* Interactively set up or change the visual representation in support of the analytical tasks. Typical visual configuration tasks include zoom, filter, and showing details on demand [4]. Much of the power of visualisation, in general, is attributed to the combination of interactive configuration and corresponding real-time updates of the graphical representation [13].

Analytical tasks. Analytical tasks are the primary building blocks for achieving a user's goal; they are the means by which specific information is obtained from the network. Analytical tasks are necessarily low-level, and applied to either individual entities or a small-subset.

– *Identify.* Find entities and/or properties in the data. At an elementary level, the identify task involves discovery of entities based on their spatial location, or based on the values of associated properties as graphically encoded in one or more visual representations. In particular, the identify task often involves finding entities in networks that are adjacent with respect to the structure of the network. The identify task can also be more involved, however, and includes the visual identification of similarities, differences, patterns, outliers, variations, relationships (proximity, dependency, independency), and uncertainty.
– *Determine.* Calculate derived properties not originally present in the data. This often involves deriving statistical measures of the properties associated with nodes and links. Examples include: sum, difference, ratio, percentile, mean, median, variance, standard deviation, correlation coefficient, and probability. In addition, the determine task includes algorithmic calculation of derived entities, for example, clustering algorithms. As the result of invoking the determine task, the visualisation is typically changed and, consequently, there is some overlap with the configure task, described above.
– *Relocate.* Revisit entities and/or properties already identified or determined. This implies that the user is already aware of the existence of these entities and/or properties, but has to exert effort to find them again. In some cases this may be trivial, but in others this may require as much effort as the initial identification task.
– *Compare.* Examine data entities and/or properties that have been identified and/or determined in contrast to each other. This often implicitly involves the relocate task. Comparisons are usually made to find similarities or differences between the properties of nodes and links. Because these properties

are visually encoded, the compare task involves contrasting spatial location and/or graphical characteristics of the visual representation of the data.

Cognitive tasks. The cognitive tasks category includes only one broad subtask: infer.

- *Infer.* Derive insight or knowledge from the data as an outcome of performing a sequence of operational and analytical tasks. The range of outcomes of an infer task is broad: it may involve forming a hypothesis, or testing a hypothesis; it may be the result of explorative analysis or serendipitous discovery; it may lead to confirmation of an expectation, to insight that contradicts expectations, or to completely new knowledge. Such knowledge may take the form of cause-and-effect relationships, trends, or probabilities.

Cognitive tasks are high-level, relate specifically to "obtaining insight" [2], and are often iteratively developed by building on prior operational, analytical, and/or cognitive tasks. The Amar and Stasko tasks, which support users in achieving high-level objectives (as outlined in the introduction), are encompassed by this category [3]. Unlike analytical tasks, cognitive tasks are often associated with uncertainty and estimation. It is possible to determine whether the result of performing an analytical task has resulted in the "correct" answer. However, cognitive tasks are more complex and tend to require significant external resources (for example, memory storage, algorithms, or computational processing time) and the notion of "accuracy" does not exist for these tasks unless such support is provided. For this reason, the unsupported execution of a cognitive task may result in an uncertain or estimated answer.

3.2 Tasks for multivariate network analysis

The task taxonomy introduced above is very general and can be applied to any data type. To meet the objective of this chapter (to describe tasks for multivariate networks), we now narrow the scope by introducing the network task taxonomy proposed by Lee et al. [7]. For network analysis, they propose four categories of tasks: *topology-based, attribute-based, browsing,* and *overview.* Lee et al.'s framework was devised by considering existing task taxonomies, by considering examples of tasks encountered in applications of network visualisation to domain problems, and by reviewing tasks involved in user studies of network visualisation methods.

Lee et al.'s framework is comprehensive in that it describes tasks commonly encountered when analysing networks. To achieve this, they propose a number of tasks for each of the categories outlined above. However, these tasks are rather node-centric in the sense that nodes are generally assumed to be the entities of interest. Consequently, although we mirror quite closely the tasks proposed by Lee et al., we have generalised these to cater also for cases where other entities, such as links, or derived entities, such as clusters, are of interest to users. We also use slightly different terminology to that originally proposed. To avoid confusion with the more restrictive meaning of "topology" in a mathematical context, we refer to the first category of tasks as "structure-based". Also, we use the term "estimation tasks" as opposed to "overview tasks" as we find that the implied meaning more closely resembles the act of imprecisely or informally gauging general network characteristics.

The premise of Lee at al.'s work is that all tasks in the categories introduced above can be considered as conjunctions of general lower-level tasks. For this, they originally used the elementary tasks proposed by Amar et al. [6]. However, we employ the tasks described in the previous section, as proposed by Valiati et al. [12], because they address some of the shortcomings of other general task taxonomies (as highlighted in the previous section). We make one exception to the approach of composing network tasks from more general tasks, however. For estimation tasks, if a precise decomposition was possible, a "correct" answer would be guaranteed, which we will argue is not the case.

Structure-based tasks. *Adjacency tasks* combine analytical tasks (identify, determine, locate, and compare) to infer knowledge about the adjacency of entities. Two entities are adjacent if there exists a path of length at most one that connects them. In most situations, once an adjacent entity has been found, the user will proceed to study a property of that entity.

Task	*Adjacency (entities)*
Description	Find the set of entities adjacent to an entity.
Examples	Find the *first names* of the **persons** directly adjacent to a **person** with the *first name* "Adam" and *last name* "Smith". Find the *types* of **relationships** directly adjacent to a **person** with the *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated) + identify/relocate *property* of **entity** (optional).

Task	*Adjacency (derived property)*
Description	Find a derived property of the entities adjacent to an entity.
Examples	Find the *number* of **persons** adjacent to a **person** with the *first name* "Adam" and *last name* "Smith". Find the *number* of **relationships** of *type* "professional" to a **person** with the *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated) + determine *derived property* of **entity**.

Task	*Adjacency (extreme properties)*
Description	Find the entity with the maximum/minimum number of adjacent entities.
Example	Find the *first name* and *last name* of the **person** with the most **relationships** of *type* “friendship”.
Decomposition	Identify/relocate **entity** with *properties* + identify/relocate adjacent **entity** + determine *derived property* of **entity** + compare *property* of **entity**.

Accessibility tasks combine analytical tasks (identify, determine, locate, and compare) to infer knowledge about the accessibility of entities. An entity is accessible from another entity if there exists a path of any length that connects them. In most situations, once an accessible entity has been found, the user will proceed to study a property of that entity.

Task	*Accessibility (entities)*
Description	Find the set of entities accessible from an entity.
Example	Find the *first names* and *last names* of the friends of friends of a **person** with the *first name* “Adam” and *last name* “Smith”. Find the *types* of **relationships** of the friends of friends of a **person** with the *first name* “Adam” and *last name* “Smith”.
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated) + identify/relocate *property* of **entity** (optional).

Task	*Accessibility (derived properties)*
Description	Find a derived property of entities accessible from an entity.
Example	Find the *number* of **persons** with direct or indirect *relationships* of **type** “managed by” to a person with the *first name* “Adam” and *last name* “Smith”.
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated) + determine *derived property* of **entity**.

Task	*Accessibility (entities, constrained)*
Description	Find the set of entities accessible from an entity where the distance is less than n.
Example	Find the *first names* and *last names* of **persons** with no more than three degrees of separation from a **person** with the *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated at most n times) + identify/relocate *property* of **entity** (optional).

Task	*Accessibility (properties, constrained)*
Description	Find a derived property of entities accessible from an entity where the distance is less than n.
Example	Find the *number* of **persons** with no more than three degrees of separation from a **person** with the *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated at most n times) + determine *derived property* of **entity**.

Common connection tasks combine analytical tasks (identify, determine, and relocate) to identify entities that share connections with two or more other entities. In most situations, once connected entities have been found, the user will proceed to study a property of those entities.

Task	*Common connection*
Description	Given a set of entities, find a set of entities that are connected to all of them.
Examples	Find the *first names* of **persons** that have direct or indirect **relationships** of *type* "managed by" to a **person** with the *first name* "Adam" and a **person** with the *first name* "Barbara". Find the *types* of direct or indirect **relationships** between a **person** with the *first name* "Adam" and a **person** with the *first name* "Barbara".
Decomposition	Identify/relocate **entity** with *property* (repeated) + identify/relocate adjacent **entity** (repeated) + determine *intersection* + identify/relocate *property* of **entity** (optional).

Connectivity tasks combine analytical tasks (identify, determine, and relocate) to infer knowledge about the connectivity of sub-networks. If *N'* is a sub-network of a network *N*, then every node and every link in *N'* is also in *N*.

Task	*Connectivity (shortest path)*
Description	Determine if two nodes are connected and find the shortest path between them.
Example	Are the **persons** with *first name* “Adam” and *first name* “Barbara” connected? Find the smallest *degree of separation* between a **person** with *first name* “Adam” and a **person** with *first name* “Barbara”.
Decomposition	Identify/relocate **entity** with *property* (repeated) + identify/relocate adjacent **entity** (repeated) + determine *derived property*.

Task	*Connectivity (clusters)*
Description	Find clusters.
Example	Identify and find the *number* of **cliques** in a social network.
Decomposition	Identify/relocate **derived entity** (repeated) + determine *derived property*.

Task	*Connectivity (connected components)*
Description	Find connected components.
Example	Identify the *number* of disconnected **sub-networks** in a social network.
Decomposition	Identify/relocate **derived entity** (repeated) + determine *derived property*.

Task	*Connectivity (bridges)*
Description	Find bridges/articulation points.
Example	Find the *first name* and *last name* of the **person** whose removal will result in a disconnected sub-network.
Decomposition	Identify/relocate **derived entity** (repeated) + identify/relocate **entity** + identify/relocate *property* of **entity**.

Attribute-based tasks. *Nodes tasks* combine analytical tasks (identify, determine, and relocate) to infer knowledge about nodes and their attributes.

Task	*Nodes (properties)*
Description	Find the nodes with specific attribute values.
Example	Find all **persons** with an *occupation* of "manager" and *age* greater than "30".
Decomposition	Identify/relocate **entity** with *property* (repeated).

Task	*Nodes (derived property)*
Description	Find a derived property of a set of nodes with specific attribute values.
Example	Find the *number* of **persons** with an *occupation* of "manager" and an *age* greater than "30".
Decomposition	Identify/relocate **entity** with *property* (repeated) + determine *derived property.*

Links tasks combine analytical tasks (identify, determine, and relocate) to infer knowledge about links and their attributes.

Task	*Links (connected nodes)*
Description	Given a node, find the nodes connected by links with specific attribute values.
Example	Find all **persons** with **relationships** of *type* "friend" to a **person** with the *first name* "Adam" and *last name* "Smith". Suppose that links are directional and that they encode managerial relationships; find all **persons** who are *managed by* a **person** with *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** with *property* (repeated).

Task	*Links (extreme values)*
Description	Find the node that is connected by a link with the minimum/maximum value for a link attribute of interest.
Example	Suppose links encode strength of friendship; find the **person** with the *strongest* friendship **relationship** with a **person** with the *first name* "Adam" and *last name* "Smith".
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** (repeated) + determine *derived property.*

Browsing tasks. *Follow path tasks* combine analytical tasks (identify and relocate) to infer knowledge about paths in multivariate networks.

Task	*Follow path*
Description	Follow a given path.
Example	Find the **person** with *first name* “Barbara” with a **relationship** of *type* “friendship” to a **person** with the *first name* of “Adam”; now find the **person** with the *first name* “Charles” with a **relationship** of *type* “friendship” to her.
Decomposition	Identify/relocate **entity** with *property* + identify/relocate adjacent **entity** with *property* (repeated).

Revisit tasks primarily employ the analytical task relocate to revisit previously visited entities. Typically this is followed with any of the other analytical tasks to infer more knowledge. Although essentially a low-level task, we include revisit here because it is part of Lee et al.’s framework and because we want to emphasise its importance in facilitating explorative analysis [7].

Task	*Revisit (entity)*
Description	Revisit an entity and infer further knowledge.
Example	After completing the previous task (follow path), go back and find the **person** with the *first name* of “Barbara” and find her other friends.
Decomposition	Relocate **entity** + identify/determine/relocate/compare.

Estimation tasks. Lee et al. propose a single “overview” task to allow estimation of general network characteristics [7]. This includes estimating the size of the network, the distribution of property values over entities, or getting a rough idea of the clusters in the network. They do not further sub-divide this category. While they state that this is a “compound exploratory task to get estimated values quickly” it is not clear how this task could be precisely decomposed into several component low-level tasks, as doing so would suggest that an exact value for the desired network characteristic could be determined (rather than an estimate, which may, of course, be inaccurate). It also suggests the use of external support in the form of memory, algorithms or computational processing time, since tasks that derive accurate characteristics of entire networks through the use of component low-level tasks can only do so if such external support is used.

Our taxonomy therefore includes “estimation” tasks. We use the term “estimation” (rather than “overview”) to emphasise that these tasks are not easily definable in terms of lower-level tasks (as per the Lee et al. definition [7]), but are high-level, with the objective of gaining a rough estimation rather than precise answers. In this sense, there is also a clear link with the “cognitive” (“infer”) task category of Valiati et al. [12], although Valiati et al., like Lee et al., suggest that these tasks can be systematically decomposed into sequences of sub-tasks.

The definition of our estimation tasks is based on the premise that external support is not available during task execution, and that precise answers are therefore not possible. The alternative would be to define overview/inference tasks algorithmically in terms of the use of low-level tasks, memory storage and computations so as to ensure accuracy; this systematic approach would add little to what is already known about task decomposition. Since neither Lee et al. nor Valiati et al. have provided sub-categories for their overview/inference tasks, we introduce categories for estimation tasks below. The sample of general information visualisation tasks defined by Amar and Stasko describe the types of high-level objectives a user may have [3] (also see the introduction), and these are used in distinguishing two types of estimation tasks: *understanding* and *comparison*.

Understanding task have the aim of gaining more complete understanding of the information; they relate to the Amar and Stasko tasks of decision making, learning and identifying domain parameters [3].

Task	*Clusters*
Description	Characterise sets of nodes as (potentially) belonging to highly-connected groups (clusters).
Example	In a social network, identify all those people who are likely to attend parties held by Adam, Barbara, and Charles. This task requires identifying a cluster of nodes for each of Adam, Barbara, and Charles. These clusters may overlap, and some nodes in the network may not belong to any of these three clusters.
Explanation	This task identifies groups of nodes that are structurally highly connected; no use is made of attribute information. The estimation is based on scanning the network structure, identifying sets of nodes that are closely linked. The definitions of the clusters may be inaccurate unless the entire network is systematically and algorithmically analysed to identify which sets of nodes form tight clusters, while keeping a record of all the connections. An estimated cluster may therefore include nodes that are only related to some (but not many) members of the cluster; or may omit some nodes that ought to be members.

Task	*Common attributes (nodes)*
Description	Characterise sets of nodes as belonging to different groups, based on node attributes.
Example	In a social network, identify all the girls who live in Glasgow, who have blue eyes, who are over 17, and who play tennis. This task is concerned with the values of five different attributes; the result is the set of nodes for which these values match the specification.
Explanation	This task identifies groups of nodes that share similar characteristics, based on several given attribute/value pairs; no use is made of structural information. The estimation is based on scanning the nodes and their attributes, identifying groups of nodes with the same attribute values. The definitions of the groups may be inaccurate unless all the nodes are systematically and algorithmically inspected to determine the values of their attributes, keeping a record of the nodes and their values. An estimated group may therefore include nodes that have only some of the correct attribute/value pairs, or may omit nodes with all the specified characteristics.

Task	*Common attributes (links)*
Description	Characterise sets of nodes as belonging to different groups, based on link attributes.
Example	In a network representing people and the email communications sent between them over the course of a week, identify all the people who sent humorous emails on Monday morning. This task is concerned with the values of the attributes associated with the links: the email content and its date.
Explanation	This task identifies groups of nodes that share similar relationships to any other nodes, based on given attribute/value pairs of their associated links. The estimation is based on scanning the nodes and their relationships (and the attributes associated with their relationships), identifying those entities associated with the correct type of relationship. The definitions of the groups may be inaccurate unless all the links are systematically and algorithmically inspected to determine the values of their attributes, keeping a record of the associated nodes. An estimated group may therefore include nodes that are not associated with the correct type of relationships, or may omit nodes that do.

Task	*Domain (nodes)*
Description	Determine the attributes and values associated with nodes.
Example	Identify all the attributes used for nodes, and all their possible values.
Explanation	This may be inaccurate unless all nodes are visited systematically or algorithmically to extract and record their attributes and values.

Task	*Domain (links)*
Description	Determine the attributes and values associated with links.
Example	Identify all the attributes used for links, and all their possible values.
Explanation	This may be inaccurate unless all links are visited systematically or algorithmically to extract and record their attributes and values.

Comparison tasks are concerned with understanding changes in a network, and relate to the Amar and Stasko tasks of identifying trends and causation, prediction, hypothesis verification, discovering correlative models, and seeing the effect of uncertainty [3]. These tasks assume the existence of more than one instance of a network, each representing a different point in time. For completeness, we include comparison tasks here, but a more detailed discussion of temporal networks is deferred to Chapter **??**.

Task	*Trends*
Description	Compare information at different stages in a changing network.
Example	In a social network, characterise how the group of friends centred around Adam changes over the course of a year.
Explanation	A changing network is described as a series of time-slices, where each time-slice is an instance of the network. This result of this task is a description of how the network has changed between two (or more) of its time-slices. Typically, it would be overview information (as described in the five "understanding" tasks above) that is compared, rather than specific node/link information. This comparison will result in uncertain information unless external algorithms are used to explicitly compare the details of the information in the series of networks.

Task	*Causation*
Description	Formulate an explanation why two time-slices in a changing network are different.
Example	Explain why some students were friends with John (the smartest student in the class) the week before an assignment was due, but not the week after.
Explanation	This task is different from the others listed above, as it requires external knowledge, that is, information that is not represented directly in the network itself.

4 Discussion

The approach that we have taken in this chapter is to review the relevant literature to come up with a pragmatic synthesis of other frameworks, with particular reference to multivariate networks. In doing so, we have considered ideas from general information visualisation methods [9], general information visualisation tasks [3], specific information visualisation questions [6], multi-dimensional visualisations [12], and visualisation tasks for univariate graphs [7].

It is worth reflecting on how the framework presented here corresponds to other recent work on information visualisation tasks. As noted before, Schulz et al. describe the visualisation task design space along five dimensions (goal, means, characteristics, target, and cardinality) [10], while Brehmer and Munzner consider three questions (why?, how?, and what?) [11]. Although these frameworks are much more general than multivariate networks and somewhat abstract for a general audience, they provide a very useful approach to reflect on some of the key points discussed in this chapter.

Goal, or *why?*, corresponds with the notion of user intent. Users study multivariate networks to gain insight about the phenomena, such as social networks, that they describe. Brehmer and Munzner emphasise that, depending on the context, there will be different levels of specificity of tasks. For example, they distinguish between high-level (consume), intermediary-level (search), and low-level (query) objectives. Schulz et al. point out that visualisation supports exploration, confirmation of hypotheses, and presentation of findings. In the light of supporting interactive analysis, our emphasis has been on the former two (exploration and confirmation). However, we note that other objectives such as presentation, communication, or even interaction with a visual representation of a data set as a form of entertainment are all valid.

Means, or *how?*, describes how a task is carried out and relates to the operational and analytical tasks described in this chapter. Most of our attention has gone into describing these tasks for a multivariate network context.

Characteristics, target, and *cardinality,* or *what?,* are concerned with how the data relates to the task. The notion of *characteristics* distinguishes between low-level and high-level aspects of the data. This corresponds closely to the difference between tasks where knowledge is directly derived from the data (for example, structure- and attribute-based tasks) and ones that require more nuanced deduction and uncertainty, as highlighted by our estimation tasks. *Target* highlights the parts of the data on which analysis focuses. This chapter picks up on this by emphasising the entities of multivariate networks (nodes, links, paths, and networks) as well as associated properties (structural properties and attributes) and how tasks relate to these. In the context of this chapter, *cardinality* emphasises that tasks may include the analysis of single or multiple networks. Although Brehmer and Munzner leave the question of *what* rather open-ended, they do emphasise the importance of defining the inputs and outputs associated with a task, especially when several tasks are combined sequentially. We do not treat this issue explicitly, but our examples imply that for our purposes the inputs are multivariate graphs and the outputs are subsets of entities (nodes, links, paths, and networks) and/or properties (structural and attributes).

Finally, it should be noted that all considered tasks necessarily involve a visual representation of one or more multivariate networks, and interaction with this visual representation. This chapter has not tried to describe interaction methods, such as filtering and zooming, which have been bundled under the configure task. We note, however, that the distinction between task and interaction method is not always clear-cut and many authors have chosen to combine examples of both (for example, [4]). Chapter **??** provides a more in-depth analysis of interaction methods for multivariate graphs.

5 Conclusion

In this chapter we have described tasks for multivariate networks. We have summarised the entities and properties of multivariate networks and presented a general taxonomy for visualisation tasks. We then described a task framework specifically for multivariate networks and showed how the proposed tasks can be composed of lower-level tasks of the general taxonomy. We also discussed some of the implications of this framework in the light of related work on information visualisation tasks.

Many of these tasks (in particular the estimation tasks) have been defined without consideration of any context or users' prior knowledge. In future work, a more semantic and situational analysis of tasks relating to multivariate networks might take into account how such knowledge might affect the way in which tasks are executed and their results interpreted. Examples of such contextual knowledge could include related node attributes (redheads tend to have blue eyes), assumed edge attributes (people tend to like their children), or broader population attributes (most computing science graduates are male).

Almost invariably, research on visualisation tasks is motivated in two ways. First, an understanding of a domain problem should be translated into user tasks to support. The user tasks, in turn, should have a direct bearing on the design of a visualisation system to address the original domain problem. Second, an understanding of user tasks enable visualisation designers to evaluate the suitability of their designs and systems in addressing a domain problem. Our aim in this chapter has been to provide an introduction and overview of tasks for multivariate network analysis to a general audience and, hence, we have not evaluated the suitability of our framework to support these objectives. We suspect that it may be useful to this end, but further work is required to make this claim.

Acknowledgements. The authors wish to thank Peter Eades (University of Sydney, Australia), Helen Gibson (Northumbria University, United Kingdom), Daniel Keim (University of Konstanz, Germany), and Robert Kosara (Tableau Software, United States) for fruitful discussions and constrictive input at the Dagstuhl Seminar *Information Visualization - Towards Multivariate Network Visualization* held at Schloss Dagstuhl, Germany, from 12 May–17 May, 2013. A.J. Pretorius was supported by a Leverhulme Early Career Fellowship under award number ECF2012-071.

References

1. Yi, J.S., Kang, Y., Stasko, J.T., Jacko, J.A.: Toward a deeper understanding of the role of interaction in information visualization. IEEE Transactions on Visualization and Computer Graphics **13**(6) (2007) 1224–1231
2. North, C.: Toward measuring visualization insight. IEEE Computer Graphics and Applications **26**(4) (2006) 6–9
3. Amar, R.A., Stasko, J.T.: Knowledge precepts for design and evaluation of information visualizations. IEEE Transactions on Visualization and Computer Graphics **11**(4) (2005) 432–442
4. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of IEEE Visual Languages. (1996) 336–343
5. Ahn, J., Plaisant, C., Shneiderman, B.: A task taxonomy of network evolution analysis. Technical Report HCIL-2012-13, Human-Computer Interaction Lab, University of Maryland (2012)
6. Amar, R.A., Eagan, J., Stasko, J.T.: Low level components of analytic activity in information visualization. In: Proceedings of the IEEE Conference on Information Visualization. (2005) 111–117
7. Lee, B., Plaisant, C., Sims Parr, C., Fekete, J.D., Henry, N.: Task taxonomy for graph visualization. In: Proceedings of the AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. (2006) 1–5
8. Shneiderman, B., Aris, A.: Network visualization by semantic substrates. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 733–740
9. Wehrend, S., Lewis, C.: A problem-oriented classification of visualization techniques. In: Proceedings of the IEEE Conference on Visualization. (1990) 139–143
10. Schulz, H.J., Nocke, T., Heitzler, M., Schumann, H.: A design space of visualization tasks. IEEE Transactions on Visualization and Computer Graphics **19**(12) (2013) 2366–2375
11. Brehmer, M., Munzner, T.: A multi-level typology of abstract visualization tasks. IEEE Transactions on Visualization and Computer Graphics **19**(12) (2013) 2376–2385
12. Valiati, E.R.A., Pimenta, M.S., Freitas, C.M.D.S.: A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In: Proceedings of the 2006 AVI workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. (2006) 1–6

13. Spence, R.: Information Visualization: Design for Interaction. Second edn. Prentice Hall (2007)