Exercises for

Chapter 11: Information Visualization

1 EXERCISE 1

Scientific visualization (scivis) and information visualization (infovis) have a related goal: understand data to draw useful conclusions. However, they treat different types of datasets. You have seen, so far, a (large) range of scivis and infovis visualization examples. Can you state that, in general, one type of visualizations is easier for you to comprehend (and use) than the other? If so, explain three reasons for which one type of visualizations is easier to understand than the other. If not, argue why the challenges provided by both types of visualization are similar or, if this is the case, of similar size.

2 EXERCISE 2

One argument in the support of the statement that scivis is different from infovis is that the former does readily allow meaningful, natural, interpolation mechanisms of the examined data (which thus allows one to easily resample scivis datasets), while the latter does often not provide such mechanisms. Give three examples of scivis datasets (and related visualization operations) and three examples of Infovis datasets (and related visualization operations) where such interpolation mechanisms exist and are crucial (for scivis data), respectively do not readily exist and would be crucial (for Infovis data).

3 EXERCISE 3

Consider a (2D or 3D) surface mesh, consisting of a set of vertices and a set of edges defining faces. Next, consider a graph drawing, consisting of a set of vertices (with 2D or 3D positions) connected by a set of edges. Fundamentally, we can argue that both types of datasets can be represented by the same *data structure* – either an unstructured grid (if we use the scivis terminology) or a graph (if we use the Infovis terminology). If so, one could argue that a graph and a mesh are one and the same thing. Give three reasons why this statement is not true.

4 EXERCISE 4

Consider the *table lens* technique used to visualize data tables (Sec. 11.3, Chapter 11). For large tables, table lenses require aggregation (or subsampling): Given $N > 1$ cell values, we can only render a single such value in the limited display space available to it (typically one color-coded pixel line). To do this, one needs a so-called *aggregation function* which takes as input the N cell values, and outputs either an 'average' cell value, next to be color-mapped to the pixel line, or the pixel-line color directly. How would you design such an aggregation function if the values were of type (a) numerical, (b) categorical, (c) text?

5 EXERCISE 5

Describe three different visualization techniques for tree datasets. For each method, describe two advantages and two disadvantages as opposed to the other methods described.

6 EXERCISE 6

In a cushion treemap, a node is rendered as a rectangle, with additional shading (to reflect its nesting or depth level in the tree) and size and color (to reflect two other attributes of the node, chosen by the user). Consider now that you have a treemap where each node has several additional attributes, say 5 or 6 in total. How can we extend the treemap method to render aspects related to the variation of several (more than two) such attributes in the limited screen space available?

7 EXERCISE 7

Graph bundling is an alternative to graph simplification for the generation of easy-to-understand visualizations of large graphs. What is a fundamental similarity between the two in terms of the tasks addressed? What is a fundamental difference?

8 EXERCISE 8

Consider the 3D triangle mesh of the surface of a cube. Here, vertices are spread densely and uniformly on all six faces of the cube, and each vertex is connected to a similar number of neighbor vertices (the mesh has a regular structure). Now, consider the adjacency graph implied by such a mesh, where each mesh vertex becomes a graph node, and each mesh triangle edge becomes a graph edge. All graph edge-weights are considered equal. What would be the expected result (shape) if we apply a good-quality 3D force-directed layout on such a graph, *e.g.* [Fruchterman and Reingold 91] or [Kamada and Kawai 89]?

9 EXERCISE 9

Graph bundling is one of the main strategies for producing visualizations of large general graphs that attempts to favor certain exploration tasks against other tasks. As opposed to *e.g.* straight-line graph drawing, graph bundling has shown to be quite effective in this respect. Which is one of the key *tasks* that graph bundling supports better than straight-line graph drawing (for the same set of vertex positions)? What is the key *graphical attribute* that bundling optimizes for to realize this effect? In contrast, which is another key task that bundling does support less well for general graphs?

10 EXERCISE 10

Dynamic graphs are graphs in which both the graph vertices and graph edges appear (and next disappear) in time. As such, the structure of such graphs can be seen as a time-dependent function. In graph visualization, two classes of methods are known for such graphs: online and offline methods. What is the fundamental difference between the two? Which class of methods is, in general, harder to implement, and why?

11 EXERCISE 11

Dynamic graphs can be classified into *sequence* graphs and *streaming* graphs. The two graph types allow different visualization methods. What is the fundamental difference between the two types of graphs?

12 EXERCISE 12

Tables and parallel coordinate plots are complementary methods for the visualization of multivariate data points. Both visualizations allocate explicit space for each value (attribute) of a data point, and both visualizations generate two-dimensional images following a Cartesian system of coordinates. What are the main differences between the two concerning their (Cartesian) layout decisions?

13 EXERCISE 13

Apart from the layout similarities (and similarities in terms of input dataset) for tables and parallel coordinate plots, a fundamental difference exists in terms of freedom of manipulating the layout to reflect different aspects of the studied data. Which is this difference?

14 EXERCISE 14

Consider a graph G consisting of N nodes and M relations (edges) having weights. Force-directed graph layouts (FDL) try to create a (2D) embedding of G so that strongly related nodes are placed close to each other. Now, consider a K-dimensional dataset D of N data points. Dimensionality reduction (DR) methods try to create a (2D) embedding of D so that strongly related data points (in terms of their K-dimensional distances) are placed close to each other. Given the similarities of the two types of techniques, how could we use a FDL method to create a DR embedding for a given K-dimensional dataset D? And how could we conversely use a DR method to create a graph layout for a given graph G?

Hints: To answer the question, first elaborate on the notation of both the graph G and dataset D. Next, define, in terms of the specific aspects of these datasets, what would be the corresponding graph G (for a K-dimensional dataset D) so that we could use a FDL on G to create a 2D embedding of D; and conversely, what would be the corresponding dataset D (for a graph G), so we could use a DR method to create a 2D embedding of G.

15 EXERCISE 15

Given a set of N K-dimensional points, dimensionality reduction methods can be classified into two main categories: Methods that use the actual dataset (*i.e.* have access to the K dimensions of all data points), called projection-based dimensionality reduction (PBDR) methods; and methods that use only the distance matrix that encodes the similarities (in K-dimensions) of the N data points, also called multidimensional scaling (MDS) methods. What is one advantage of PBDR methods *vs* MDS methods? What is one disadvantage of PBDR methods *vs* MDS methods?

16 EXERCISE 16

Consider a 2D point-cloud created by a dimensionality reduction (DR) method, where each 2D point represents a K-dimensional data point. We assume here that closely-placed 2D points reflect data points which are highly similar in K dimensions. One major challenge of using such DR plots is to understand why points have been placed close to each other, and what do compact groups of 2D points mean in the final 2D plot. Describe two visualization methods (*e.g.* in terms of mapping or interaction techniques) that address the above two tasks.

17 EXERCISE 17

Consider a K-dimensional dataset of N points having real-valued attributes. For this dataset, we use a similarity function being the K-dimensional Euclidean distance metric. We project this dataset in 2D using one of the existing dimensionality reduction (DR) methods available. We call the resulting 2D projection D_1. Next, we perform two operations: First, we uniformly scale all K dimensions of our dataset by some value w ($w \neq 0, w \neq 1$), and reproject, to obtain a 2D projection called D_2. Secondly, we scale only one of the K dimensions of our dataset by the value w, and reproject, to obtain the 2D projection called D_3. Assuming that our DR method is well suited to preserve K-dimensional neighborhoods, based on an Euclidean distance metric, what can we say about the resulting projections D_1, D_2, and D_3:

- D_1 should be visually identical to D_2, but not identical to D_3
- D_1 should be visually identical to D_3, but not identical to D_2
- D_1, D_2, and D_3 are all visually identical

18 EXERCISE 18

Consider a given dataset of N K-dimensional points which we project in 2D twice, by using *e.g.* two different dimensionality reduction (DR) methods, or the same method run twice with different parameters. We thus obtain two 2D projections, or point clouds which we call D_1 and D_2. In general, D_1 and D_2 will not be identical. We assume that both D_1 and D_2 are rendered as 2D point clouds, either overlapped to each other, or in separate views. Describe a simple visualization method that highlights the main similarities and/or differences between D_1 and D_2, without having to resort to user interaction. In particular, your method should be able to highlight

- which individual points have (nearly) the same locations in D_1 and D_2
- which individual points have significantly different locations in D_1 and D_2
- which *groups* of points, defined as sets of points which are closely placed to each other, have significantly different locations in D_1 and D_2

19 EXERCISE 19

Consider the development of a plagiarism-detection tool. As input, this tool takes two text documents, f_1 and f_2, *e.g.* PDF files or plain-text files. Next, the tool detects blocks of text which are (nearly) identical in the two documents. As output, the tool delivers a list of so-called duplicates $B = \{(b_i^1, b_i^2, w_i)\}$. A duplicate is a tuple of three elements: b_i^1 is a text-block in f_1 which was found to be very similar to text-block b_i^2 in f_2; the similarity between b_i^1 and b_i^2 is encoded by a positive value w_i, where larger values indicate more similar blocks.

We would like to visualize the duplicate set B in a way that allows us to easily and quickly complete several tasks. In this context, answer the following questions

- How would you visually present the information stored in B so that the user quickly sees which parts of the two documents contain many, few, or no duplicates?
- How would you visually present the information stored in B so that the user quickly sees where, in one of the two documents, a specific block in the other document is duplicated?
- Discuss briefly how you would extend your proposed visualization(s) to handle both above questions if we have, as input, a set of *three* documents

Support your explanations, where needed, by sketches of the proposed visualizations.

End of Exercises for Chapter 11: Information Visualization