CSE 564
Visualization & Visual Analytics

Applications and Basic Tasks

Klaus Mueller

Computer Science Department
Stony Brook University
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Topic</th>
<th>Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Applications of visual analytics, basic tasks, data types</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Introduction to D3, basic vis techniques for non-spatial data</td>
<td>Project #1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data assimilation and preparation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bias in visualization</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Data reduction and dimension reduction</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Visual perception and cognition</td>
<td>Project #1 due</td>
</tr>
<tr>
<td>8</td>
<td>Visual design and aesthetics</td>
<td>Project #2 out</td>
</tr>
<tr>
<td>9</td>
<td>Python/Flask hands-on</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cluster analysis: numerical data</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster analysis: categorical data</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Foundations of scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Computer graphics and volume rendering</td>
<td>Project #2 due / Project #3 out</td>
</tr>
<tr>
<td>14</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Illustrative rendering</td>
<td>Project #3 due</td>
</tr>
<tr>
<td>16</td>
<td>High-dimensional data, dimensionality reduction</td>
<td>Final project proposal call out</td>
</tr>
<tr>
<td>17</td>
<td>Correlation visualization</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Midterm #1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Visual analytics and the visual sense making process</td>
<td>Final project proposal due</td>
</tr>
<tr>
<td>21</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Visualization of time-varying and time-series data</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of streaming data</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of graph data</td>
<td>Final Project preliminary report due</td>
</tr>
<tr>
<td>25</td>
<td>Visualization of text data</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Midterm #2</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Data journalism</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Final project presentations</td>
<td>Final Project slides and final report due</td>
</tr>
</tbody>
</table>
Data Types Every CS Person Knows

Data type

- Primitive
  - char
  - int
  - float
  - double
  - void
- Derived
  - Array
  - Pointer
  - Function
- User defined
  - enum
  - Structure
  - Union
**Data Types in Visual Analytics**

- Numeric
- Categorical
- Text
- Time series
- Graphs and networks
- Hierarchies
Variables in Statistics

Numeric variables
- measure a **quantity** as a number
- like: ‘how many’ or ‘how much’
- can be continuous (grey curve)
- or discrete (red steps)

Categorical variables
- describe a **quality** or characteristic
- like: ‘what type’ or ‘which category’
Most often the x-axis is ‘time’

- provides an intuitive & innate ordering of the data values
- the majority of people expect the x-axis to be ‘time’

But ‘time’ is not the only option

- engineers, statisticians, etc. will be receptive to this idea
- can you think of an example?
Another plot where ‘time’ is not the x-axis

- from the engineering / physics domain
- in some sense, it tells a story
Variables in Statistics

Numeric variables
- measure a **quantity** as a number
- like: ‘how many’ or ‘how much’
- can be continuous (grey curve)
- or discrete (red steps)

Categorical variables
- describe a **quality** or characteristic
- like: ‘what type’ or ‘which category’
- can be ordinal = ordered, ranked (distances need not be equal)
  - clothing size, academic grades, levels of agreement
- or nominal = not organized into a logical sequence
  - gender, business type, eye color, brand
CATEGORICAL VARIABLES

Usually plotted as bar charts or pie charts

nominal

ordinal

but of course you can plot either of them in either of these two representations
But not everything is expressed in numbers
- images
- video
- text
- web logs
- ...

Do **feature analysis** to turn these abstract things into numbers
- then apply your analysis as usual
- but keep the reference to the original data so you can return to the native domain where the analysis problem originated
Characteristics
- often large scale
- time series

Feature Analysis
- example: Motif discovery
- encode into 5D data vector

% features discovered in stream
[0.12, 0.3, 0.41, 0.12, 0.05]
[feat. 1, feat. 2, .., feat. 5]
Sensor Data

Characteristics
- often large scale
- time series

Feature Analysis
- Fourier transform (FT, FFT)
- Wavelet transform (WT, FWT)

Fourier transform
Characteristics

- array of pixels

Feature Analysis

- value histograms
- encode into a 256-D vector

histograms

[0, 0, 0, ..., 10, ..., 1200, .....]
Characteristics

- array of pixels

Feature Analysis

- value histograms
- gradient histograms
- FFT, FWT
- Scale Invariant Feature Transform (SIFT)
- Bag of Features (BoF)
- visual words
Bag of Features (BoF)
**Bag of Features (BoF)**

1. Obtain the set of bags of features
   (i) Select a large set of images
   (ii) Extract the SIFT feature points of all the images in the set and obtain the SIFT descriptor for each feature point extracted from each image
   (iii) Cluster the set of feature descriptors for the amount of bags we defined and train the bags with clustered feature descriptors
   (iv) Obtain the visual vocabulary

2. Obtain the BoF descriptor for a given image/video frame
   (v) Extract SIFT feature points of the given image
   (vi) Obtain SIFT descriptor for each feature point
   (vii) Match the feature descriptors with the vocabulary we created in the first step
   (viii) Build the histogram
Characteristics

- essentially a time series of images

Feature Analysis

- many of the above techniques apply albeit extension is non-trivial
Characteristics

- often raw and unstructured

Feature analysis

- first step is to remove stop words and stem the data
- perform **named-entity recognition** to gain atomic elements
  - identify names, locations, actions, numeric quantities, relations
  - understand the structure of the sentence and complex events
- example:
  - Jim bought 300 shares of Acme Corp. in 2006.
  - \([\text{Jim}]_{\text{Person}} \text{ bought } [300 \text{ shares}]_{\text{Quantity}} \text{ of } [\text{Acme Corp.}]_{\text{Organiz.}} \text{ in } [2006]_{\text{Time}}\)
- distinguish between
  - application of grammar rules (old style, need experienced linguists)
  - statistical models (Google etc., need big data to build)
Create a term-document matrix

- turns text into a high-dimensional vector which can be compared
- use Latent Semantic Analysis (LSA) to derive a visualization

<table>
<thead>
<tr>
<th>Index Words</th>
<th>book</th>
<th>dads</th>
<th>dummies</th>
<th>estate</th>
<th>guide</th>
<th>investing</th>
<th>market</th>
<th>real</th>
<th>rich</th>
<th>stock</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T9</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Term-Document Matrix

Word/document cluster

---

LSA

XY Plot of Words and Titles
Train a shallow neural network (NN) on a corpus of text

- the NN weight vectors encode word similarity as a high-D vector
- use a 2D embedding technique to display
Load up the word vectors

QUEEN [0.3, 0.9]
KING [0.5, 0.7]
WOMAN [0.3, 0.4]
MAN [0.5, 0.2]

gender = WOMAN – MAN
QUEEN = KING + gender

QUEEN = KING – MAN + WOMAN
Word Cloud

Maps the frequency of words in a corpus to size

https://www.jasondavies.com/wordcloud/
Weblogs
- typically represented as text strings in a pre-specified format
- this makes it easy to convert them into multidimensional representation of categorical and numeric attributes

Network traffic
- characteristics of the network packets are used to analyze intrusions or other interesting activity
- a variety of features may be extracted from these packets
  - the number of bytes transferred
  - the network protocol used
  - IP ports used
Let’s Look at Some Essential Graphical Representations

And Do Some Advertising for D3
Stakeholder Hierarchy

- Stakeholders
  - Customers
    - Favored User Classes
  - Others
    - Disfavored User Classes
    - Ignored User Classes
    - Other User Classes
  - Users
FUNCTION CALL TREE

gprof_function_call_tree_view_get_type
  gprof_function_call_tree_view_init
  gprof_function_call_tree_view_get_type
  gprof_function_call_tree_view_init
  gprof_function_call_tree_view_class_init
  gprof_function_call_tree_view_refresh
  gprof_function_call_tree_view_add_function
  gprof_call_graph_block_get_next
  gprof_call_graph_block_is_recursive
  gprof_call_graph_block_get_first_child
  gprof_call_graph_find_block
  gprof_call_graph_block_entry_get_next
  gprof_call_graph_block_entry_get_name
  gprof_call_graph_block_get_first_block
  gprof_profile_data_get_call_graph
  gprof_view_get_data
  gprof_call_graph_block_get_primary_entry
  gprof_call_graph_block_get_next
  gprof_function_call_tree_view_finalize
  gprof_function_call_tree_view_get_widget
  gprof_function_call_tree_view_create_columns
  on_list_view_row_activated
  gprof_view_show_symbol_in_editor
Questions you might have

- how large is each group of stakeholders (or function)?
  - tree with quantities
- what fraction is each group with respect to the entire group?
  - partition of unity
- how is information disseminated among the stakeholders (or functions)?
  - information flow
- how close (or distant) are the individual stakeholders (functions) in terms of some metric?
  - force directed layout
More scalable tree, and natural with some randomness

http://animateddata.co.uk/lab/d3-tree/
A standard tree, but one that is scalable to large hierarchies

Zoomable Partition Layout

A tree that is scalable and has partial partition of unity

More space efficient since it’s radial, has partial partition of unity

http://bl.ocks.org/kerryrodden/7090426
Bubble Charts

No hierarchy information, just quantities

http://bl.ocks.org/mbostock/4063269
Quantities and containment, but not partition of unity

Quantities, containment, and full partition of unity

Relationships among group fractions, not necessarily a tree

http://bl.ocks.org/mbostock/4062006
Hierarchical Edge Bundling

Relationships of individual group members, also in terms of quantitative measures such as information flow

Collapsible Force Layout

Relationships within organization members expressed as distance and proximity

Voronoi Tessellation

Shows the closest point on the plane for a given set of points... and a new point via interaction

http://bl.ocks.org/mbostock/4060366
Data Type Conversions and Transformation
Solution 1:

- divide the numeric attribute values into $\varphi$ equi-width ranges
- each range/bucket has the same width
- example: customer age

- what is lost here?
Age ranges of customers could be unevenly distributed within a bin
- this could be an interesting anomaly
Solution 2:

- divide the numeric attribute values into $\varphi$ **equi-depth** ranges
- same number of samples in each bin
- (again) example: customer age:

![Bar chart]

- what is the disadvantage here?
- extra storage needed: must store the start/end value for each bin
Solution 3:

- what if all the bars have seemingly the same height
- or are dominated by one large peak

- switch to log scaling of the y-value
Dang and Wilkinson, “Transforming Scagnostics to Reveal Hidden Features”, TVCG 2014

Other Transformations

- **None**: $x^* = x$ (leaves points unchanged)
- **Half**: $x^* = x/2$ (squeezes all points together)
- **Square**: $x^* = x^2$ (pulls points toward left of frame)
- **Square root**: $x^* = \sqrt{x}$ (mildly pulls points toward right of frame)
- **Log**: $x^* = \log(x)$ (strongly pulls points toward right of frame)
- **Inverse**: $x^* = 1/x$ (reverses scale and squeezes points into left of frame)
- **Logit**: $x^* = (\log(x/(1-x)) + 10)/20$ (squeezes points toward middle of frame)
- **Sigmoid**: $x^* = 1/(1+\exp(-20x+10))$ (expands points away from middle of frame)
Ever tried to reduce the size of an image and you got this?

This is aliasing
But what you really wanted is this:

This is *anti-aliasing*
The smaller image resolution cannot represent the image detail captured at the higher resolution

- skipping this small detail leads to these undesired artifacts
Procedure

- either sample at a higher rate
- or smooth the signal before sampling it
- the latter is called *filtering*
ANTI-ALIASING VIA SMOOTHING

X360

PS3
Anti-Aliasing Via Smoothing
What is Smoothing?

Slide a window across the signal

- stop at each discrete sample point
- average the original data points that fall into the window
- store this average value at the sample point
- move the window to the next sample point
- repeat

[Diagram showing a signal with raw data and smoothed data]
Anti-Aliasing Via Smoothing: Tradeoffs

Looks sharper, but has "jaggies"  a bit blurred, but no more jaggies
What is the filter we just used called?

- it’s called a *box filter*

There are other filters

- for example, Gaussian filter
- yields a smoother result
- box filtering is simplest
Box Filter vs. Gaussian Filter

Can you see some patterns?

It’s another form of aliasing
What’s the underlying problem?
- detail can’t be refined upon zoom
- can just be replicated or blurred

The solution...
- represent detail as a function that can be mathematically refined
- replace raster graphics by vector graphics
Scalable Vector Graphics (SVG)
Vector graphics tends to have an “cartoonish” look

raster graphics  vector graphics
PHOTOGRAPHS AND IMAGES IN SVG
D3 Uses SVG

The Wealth & Health of Nations

1835
Filtering/smoothing also eliminates noise in the data
In some ways, bar charts reduce noise and uncertainties in the data
  - the bins do the smoothing

Example:
  - obesity over age (group)
Of course, bar charts can also hold categorical data

- smoothing by semantic grouping
- for example, Europe vs. {France, Spain, Italy, Germany, ...}
Bar Charts vs. Histograms

Histograms
- bars show the frequency of numerical data
- quantitative data
- elements are grouped together, so that they are considered as ranges
- bars cannot be reordered
- width of bars need not be the same

Bar charts
- uses bars to compare different categories of data
- comparison of discrete variables
- elements are taken as individual entities
- bars can be reordered
- width of bars need to be the same
How many bars are too many (in a chart)

- if individual categories are the focus? 12 is a good rule
- if the overall trend is the important factor? 50 or even more
- eventually you can switch to a line chart

- sort bars by height and use ‘other’ to aggregate the bar chart tails into a single bar
- find a grouping that can semantically aggregate bars, for example aggregate countries into continents

more information
BAR CHARTS IN D3

http://bl.ocks.org/mbostock/3885304

Working with bar charts and histograms is the topic of Lab 1
  ▪ the next two slides offer some help with calculations
Determine bin size

- min(data) is optional, can also use 0 or some reasonable value
- max(data) is optional, can also use some reasonable value

\[
\text{bin size} = \frac{\text{max(data)} - \text{min(data)}}{\text{number of bins}}
\]

Given a data value \( \text{val} \) increment (++) the bin value

- but first initialize bin val array to 0

\[
\text{bin val array} \left[ \left\lfloor \frac{\text{val} - \text{min(data)}}{\text{bin size}} \right\rfloor \right] + +
\]
**Histogram Calculations – Plotting**

Determine bin size on the screen

\[ \text{bin size on screen} = \frac{\text{chart width}}{\text{number of bins}} \]

Center of a bar for bin with index \( \text{bin index} \)

\[ \text{bar center on screen} = (\text{bin index} \cdot \text{bin size on screen}) + 0.5 \]

Height of the bar for a bin with index \( \text{bin index} \)

\[ \text{bar height}(\text{bin index}) = \text{bin val array}(\text{bin index}) \cdot \frac{\text{chart height}}{\max(\text{bin val array})} \]

Do not forget that the origin of a web page is the top left corner.