CSE 564
Visualization & Visual Analytics
Applications and Basic Tasks

Klaus Mueller
Computer Science Department
Stony Brook University
http://www.cs.stonybrook.edu/~mueller/teaching/cse564/

Everything you need is there:
- syllabus
- course notes (slides) posted shortly after the lecture
- lab assignments
- course policy

There will also be (soon to be announced)
- a server for lab assignments
- piazza for online support
<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>Visual perception and cognition</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Visual design and aesthetics</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Data types, notion of similarity and distance</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Data preparation and reduction</td>
<td>Project #1 due</td>
</tr>
<tr>
<td>8</td>
<td>Introduction to R, statistics foundations</td>
<td>Project #2 out</td>
</tr>
<tr>
<td>9</td>
<td>Data mining techniques: clusters, text, patterns, classifiers</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Data mining techniques: clusters, text, patterns, classifiers</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Computer graphics and volume rendering</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Techniques to visualize spatial (3D) data</td>
<td>Project #2 due</td>
</tr>
<tr>
<td>13</td>
<td>Scientific and medical visualization</td>
<td>Project #3 out</td>
</tr>
<tr>
<td>14</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Midterm #1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>High-dimensional data, dimensionality reduction</td>
<td>Project #3 due</td>
</tr>
<tr>
<td>17</td>
<td>Big data: data reduction, summarization</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Correlation and causal modeling</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Principles of interaction</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
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
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
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]
Characteristics
- often large scale
- time series

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

Fourier transform
**Image Data**

**Characteristics**
- An array of pixels

**Feature Analysis**
- Value histograms
- Encode into a 256-D vector

```
[0, 0, 0, ..., 10, ..., 1200, .....]
```
**IMAGE DATA**

**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)
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
VIDEO DATA

Characteristics
- essentially a time series of images

Feature Analysis
- many of the above techniques apply albeit extension is non-trivial
Text Data

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.
  - [Jim]_{Person} bought [300 shares]_{Quantity} of [Acme Corp.]_{Organiz.} in [2006]_{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>Titles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1</td>
</tr>
<tr>
<td>book</td>
<td>1</td>
</tr>
<tr>
<td>dads</td>
<td></td>
</tr>
<tr>
<td>dummies</td>
<td>1</td>
</tr>
<tr>
<td>estate</td>
<td></td>
</tr>
<tr>
<td>guide</td>
<td>1</td>
</tr>
<tr>
<td>investing</td>
<td>1</td>
</tr>
<tr>
<td>market</td>
<td>1</td>
</tr>
<tr>
<td>real</td>
<td></td>
</tr>
<tr>
<td>rich</td>
<td></td>
</tr>
<tr>
<td>stock</td>
<td>1</td>
</tr>
<tr>
<td>value</td>
<td></td>
</tr>
</tbody>
</table>

Term-Document Matrix

Word/document cluster

LSA
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
**WORD EMBEDDING ALGEBRA**

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**
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
  - Others
    - Users
      - Favored User Classes
      - Disfavored User Classes
      - Ignored User Classes
      - Other User Classes

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/
Collapsible Tree

A standard tree, but one that is scalable to large hierarchies

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
Circle Packing

Quantities and containment, but not partition of unity

Quantities, containment, and full partition of unity

Chord Diagram

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

Relationships within organization members expressed as distance and proximity

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:

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 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
Why discrete?

- because we can’t store continuous data
- we can only store samples of the continuous data
- how many samples do we need?
- also keep this in mind for data reduction
Why discrete?

- because we can’t store continuous data
- we can only store samples of the continuous data
- how many samples do we need?

sample points $x[n]$

original signal $x_c$
Why discrete?

- because we can’t store continuous data
- we can only store samples of the continuous data
- how many samples do we need?
Why discrete?

- because we can’t store continuous data
- we can only store samples of the continuous data
- how many samples do we need?

We need a certain number of samples to represent a continuous phenomenon

- twice as many samples as the highest frequency in the signal
- called the *Nyquist frequency*
- else we get *aliasing*
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*
Why Is This Happening?

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
Anti-Aliasing Via 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
ANTI-ALIASING VIA SMOOTHING: TRADEOFFS

looks sharper, but has “jaggies”
da 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
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
Photographs and Images in SVG
D3 Uses SVG

The Wealth & Health of Nations

42.0%
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, \ldots\}
BAR CHARTS IN D3

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

Working with bar charts will be your job for 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

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

Given a data value \( val \) increment (++) the bin value
- but first initialize bin val array to 0

\[
bin\ val\ array\left\lceil \frac{val - \text{min}(data)}{bin\ size} \right\rceil + +
\]
Determine bin size on the screen

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

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

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

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

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

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