Ch 13: Reduce Items and Attributes
Papers: Glimmer

Tamara Munzner
Department of Computer Science
University of British Columbia

CPSC 547, Information Visualization
Day 13: 3 November 2015

http://www.cs.ubc.ca/~tmm/courses/547-15
News

• marks for pitches and Q12 not ready yet

• reminder: meetings due by Thu 5pm
• reminder: proposals due by Mon 5pm
• topic requests were due yesterday
Idiom design choices: Part 2

<table>
<thead>
<tr>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change</td>
<td>Juxtapose</td>
<td>Filter</td>
</tr>
<tr>
<td>Select</td>
<td>Partition</td>
<td>Aggregate</td>
</tr>
<tr>
<td>Navigate</td>
<td>Superimpose</td>
<td>Embed</td>
</tr>
</tbody>
</table>
Reduce items and attributes

- reduce/increase: inverses

- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind

- aggregation
  - pro: inform about whole set
  - con: difficult to avoid losing signal

- not mutually exclusive
  - combine filter, aggregate
  - combine reduce, change, facet

Reducing Items and Attributes

- Filter
  - Items
  - Attributes

- Aggregate
  - Items
  - Attributes
Idiom: dynamic filtering

- item filtering
- browse through tightly coupled interaction
  - alternative to queries that might return far too many or too few

System: FilmFinder

Idiom: scented widgets

• augment widgets for filtering to show information scent
  – cues to show whether value in drilling down further vs looking elsewhere
• concise, in part of screen normally considered control panel

Idiom: **DOSFA**

- attribute filtering
- encoding: star glyphs

Idiom: **histogram**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly
Idiom: **boxplot**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower and upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown

[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]
Idiom: **Hierarchical parallel coordinates**

- dynamic item aggregation
- derived data: **hierarchical clustering**
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy

Dimensionality reduction

• attribute aggregation
  – derive low-dimensional target space from high-dimensional measured space
  – use when you can’t directly measure what you care about
    • true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    • latent factors, hidden variables

Tumor Measurement Data

data: 9D measured space

\[ \text{DR} \]

derived data: 2D target space

Malignant

Benign
Dimensionality reduction for documents

**Task 1**

*In* HD data → *Out* 2D data

- **What?**
  - In High-dimensional data
  - Out 2D data
- **Why?**
  - Produce
  - Derive

**Task 2**

*In* 2D data → *Out* Scatterplot Clusters & points

- **What?**
  - In 2D data
  - Out Scatterplot
  - Out Clusters & points
- **Why?**
  - Discover
  - Explore
  - Identify
- **How?**
  - Encode
  - Navigate
  - Select

**Task 3**

*In* Scatterplot Clusters & points → *Out* Labels for clusters

- **What?**
  - In Scatterplot Clusters & points
  - Out Labels for clusters
- **Why?**
  - Produce
  - Annotate

*Wombat*
Dimensionality vs attribute reduction

• vocab use in field not consistent
  – dimension/attribute

• attribute reduction: reduce set with filtering
  – includes orthographic projection

• dimensionality reduction: create smaller set of new dims/attrs
  – typically implies dimensional aggregation, not just filtering
  – vocab: projection/mapping
Estimating true dimensionality

• how do you know when you would benefit from DR?
  – consider error for low-dim projection vs high-dim projection

• no single correct answer; many metrics proposed
  – cumulative variance that is not accounted for
  – strain: match variations in distance (vs actual distance values)
  – stress: difference between interpoint distances in high and low dims

\[
\text{stress}(D, \Delta) = \sqrt{\frac{\sum_{ij} (d_{ij} - \delta_{ij})^2}{\sum_{ij} \delta_{ij}^2}}
\]

- **D**: matrix of lowD distances
- **\Delta**: matrix of hiD distances \( \delta_{ij} \)
Estimating true dimensionality

• scree plots as simple way: error against # attrs

– original dataset: 294 dims
– estimate: almost all variance preserved with < 20 dims

[Fig 2. DimStiller: Workflows for dimensional analysis and reduction. Ingram et al. Proc. VAST 2010, p 3-10]
Dimensionality Reduction

• why do people do DR?
  – improve performance of downstream algorithm
    • avoid curse of dimensionality
  – data analysis
    • if look at the output: visual data analysis!

• DR tasks
  – dimension-oriented task sequences
    • name synthetic dimensions, map synthetic dims to original ones
  – cluster-oriented task sequences
    • verify clusters, name clusters, match clusters and classes

Linear dimensionality reduction

- principal components analysis (PCA)
  - describe location of each point as linear combination of weights for each axis
  - finding axes: first with most variance, second with next most, ...
Nonlinear dimensionality reduction

• many techniques proposed
  – MDS, charting, isomap, LLE, T-SNE
  – many literatures: visualization, machine learning, optimization, psychology, ...

• pro: can handle curved rather than linear structure

• cons: lose all ties to original dims/attrs
  – new dimensions cannot be easily related to originals
MDS: Multidimensional Scaling

• confusingly: entire family of methods, linear and nonlinear!

• classical scaling: minimize strain
  – early formulation equivalent to PCA (linear)
  – Nystrom/spectral methods approximate eigenvectors: \( \mathcal{O}(N) \)
    • Landmark MDS [de Silva 2004], PivotMDS [Brandes & Pich 2006]
  – limitations: quality for very high dimensional sparse data

• distance scaling: minimize stress
  – nonlinear optimization: \( \mathcal{O}(N^2) \)
    • SMACOF [de Leeuw 1977]
  – force-directed placement: \( \mathcal{O}(N^2) \)
    • Stochastic Force [Chalmers 1996]
    • limitations: quality problems from local minima

• Glimmer goal: \( \mathcal{O}(N) \) speed and high quality
Spring-based MDS: naive

• repeat for all points
  – compute spring force to all other points
  – difference between high dim, low dim distance
  – move to better location using computed forces

• compute distances between all points
  – $O(N^2)$ iteration, $O(N^3)$ algorithm
Faster spring model: Stochastic

• compare distances only with a few points
  – maintain small local neighborhood set
  – each time pick some randoms, swap in if closer

• small constant: 6 locals, 3 randoms (typically)
  – $O(N)$ iteration, $O(N^2)$ algorithm
Faster spring model: Stochastic

• compare distances only with a few points
  – maintain small local neighborhood set
Glimmer algorithm

- multilevel to avoid local minima, designed to exploit GPU
- restriction to decimate
- relaxation as core computation
- relaxation to interpolate up to next level

Glimmer Strategy

• stochastic force alg suitable for fast GPU port
  – but systematic testing shows it often terminates too soon

• use as subsystem within new multilevel GPU alg with much better convergence properties

[Fig 2,4. Glimmer: Multilevel MDS on the GPU. Ingram, Munzner, Olano. IEEE TVCG 15(2):249-261, 2009.]
Stochastic termination

• how do you know when it’s done?
  – no absolute threshold, depends on the dataset
  – interactive click to stop does not work for subsystem

• sparse normalized stress approximation
  – minimal overhead to compute (vs full stress)
  – low pass filter

GPUs

• characteristics
  – small set of localized texture accesses
  – output at predetermined locations
  – no variable length looping
  – avoid conditionals: all floating point units execute same instr at same time

• mapping problems to GPU
  – arrays become textures
  – inner loops become fragment shader code
  – program execution becomes rendering
Finding and verifying clusters

- sparse docs dataset
  - 28K dims, 28K points
    - speed equivalent to classical
    - quality major improvement

[Fig 8, 9. Glimmer: Multilevel MDS on the GPU. Ingram, Munzner, Olano. IEEE TVCG 15(2):249-261, 2009.]
Methods and outcomes

• methods
  – quantitative algorithm benchmarks: speed, quality
    • systematic comparison across 1K-10K instances vs a few spot checks
  – qualitative judgements of layout quality

• outcomes
  – characterized kinds of datasets where technique yields quality improvements
    • sparse documents

• followup work
  – Q-SNE: millions of documents

Next Time

• meetings: by 5pm Thu
  – I’m gone Fri and Mon

• proposals: by 5pm Mon

• Thu Nov 5, to read
  – VAD Ch. 14: Embed Focus+Context