Lecture 10: Attribute Reduction Methods
Information Visualization
CPSC 533C, Fall 2011

Tamara Munzner
UBC Computer Science

Wed, 12 October 2011
Chapter 8: Attribute Reduction Methods

Further Reading


Data Reduction

- how to reduce amount of stuff to draw?
  - crosscuts view composition considerations

- item reduction
  - last time
  - rows of table

- attribute reduction
  - this time
  - columns of table

- methods for both
  - filtering, aggregation, ordering
Attribute Reduction Methods

- camera metaphors
  - slicing, cutting, projection
- filtering, ordering, aggregation
  - for attributes as opposed to items
- dimensionality reduction
  - uncovering hidden structure
  - estimating true dimensionality
  - generating synthetic dimensions
    - linear mappings
    - nonlinear mappings
- displaying low-dimensional spaces
  - scatterplots, SPLOMS, landscapes
Slicing/Cutting: Spatial Data

- easy to understand: spatial data, 3D to 2D, axis aligned

Slicing: High-Dimensional Functions

- HyperSlice: matrix of orthogonal 2D slices
- each panel is display and control: drag to change slice
- simple 3D example

Figure 2: Effect of dragging a slice

The values along a horizontal line through the center of the panel are the same for all panels in the same column, and also the values along a vertical line are the same for all panels in the same row (see fig. 1).

This HyperSlice representation allows the viewer to observe the sensitivity of the function to changes in one and two dimensions. It is difficult, if not impossible, to reconstruct a complete, multi-dimensional mental image from the separate graphical representations. However, this representation does enable the user to view the multi-dimensional space around a point in a simple and intuitive way. The user can locate features such as extrema and hyperplanes. Because all dimensions are presented simultaneously and in various combinations, the chance that important relations are overlooked is small. Another interesting property is that for the HyperSlice reduces to the standard representation: a single graph.

The main strength of the HyperSlice representation is that it lends itself very well to interaction via direct manipulation, which is the subject of the next section.

3 Interaction

3.1 Navigation

The HyperSlice representation shows only around the current point \( c \). Probably the most important aspect of user interaction is therefore the change of \( c \). By changing \( c \) the user steers through multi-dimensional space in search for interesting features of the function, where the visual representation supports his navigation. A direct and simple solution is feasible with the HyperSlice concept. The user can point at a panel, press a mouse-button, and drag the visual representation. If the user drags a slice over a displacement \( \Delta \), then the current point is changed as follows:

\[
\text{[Equation]} \quad c \rightarrow c + \Delta
\]

The visual effect is shown in figure 2. Here the slice is dragged. Slices in the same column move horizontally over a displacement \( \Delta \), whereas the slices in the same row move vertically over a displacement \( \Delta \). Furthermore, for all slices other than \( \xi \), one or two of the modified dimensions of the current point are not represented by a horizontal or vertical axis. One could say that these dimensions are perpendicular to these slices. A change in such a dimension does affect the slice shown: the slices move perpendicular to the image plane.

If the graph is dragged, the single variable is changed. The effect is similar to that as described for slices. Thus, each panel serves not only as a visual representation, but also as one- or two-dimensional sliders for the current value of variables.

In practice this mechanism is used in various ways: If in one of the panels an interesting spot is detected (e.g. an optimum) the user can drag this spot to the center of the panel;

[Fig 1, 2. van Wijk and van Liere. HyperSlice: Visualization of scalar functions of many variables. Proc. IEEE Visualization 1993]
Slicing: HyperSlice

- 4D function \( \sum_{i=0}^{3} w_i/(1 + |x - p_i|^2) \)
- diagonals = standard graph

[Fig 4. van Wijk and van Liere. HyperSlice: Visualization of scalar functions of many variables. Proc. IEEE Visualization 1993]
Slicing: HyperSlice

- satellite orbit eccentricity: x pos, y pos, x vel, grav const

[Fig 4. van Liere and van Wijk. Visualization of Multi-Dimensional Scalar Functions Using HyperSlice. CWI Quarterly, 7(2), June 1994, 147-158.]
Projections

- orthographic: remove all information about filtered dims
- hypercube: 3D to 2D, 4D to 3D (video)
- perspective: some info about filtered dims remains

Attribute Filtering

- filtering, but for attributes rather than items
- unfiltered vs filtered SPLOM

[Fig 4. Yang et al. Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration Of High Dimensional Datasets. Proc. InfoVis 2003]
Attribute Ordering

- ordering, but for attributes rather than items
- Hierarchical Clustering Explorer

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
  - set size is smaller than original, new dims completely synthetic
  - clarification: includes dimensional aggregation
  - includes some projections (but not all)
    - vocab: projection/mapping
Uncovering Hidden Structure

- measurements indirect not direct
  - real-world sensor limitations
- measurements made in sprawling space
  - documents, images
- DR only suitable if (almost) all information could be conveyed with fewer dimensions
  - how do you know? need to estimate true dimensionality to check if different than original!
Estimating True Dimensionality

- error for low-dim projection vs high-dim original
- 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 dimensions

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

- \(D\): matrix of lowD distances
- \(\Delta\): matrix of hiD distances \(\delta_{ij}\)
Showing Dimensionality Estimates

- scree plots as simple way: error against \# dims
  - original dataset: 294 dims
  - estimate: almost all variance preserved with < 20 dims

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

- principal components analysis
  - 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, TSNE,...
  - optimization problem
- pro: can handle curved rather than linear structure
- con: lose all ties to original dimensions
  - new dimensions cannot be easily related to originals
DR in Visualization: Tasks

- find/verify new/synthetic dimensions
  - are the new dimensions believable?
  - ex: data-driven reflectance model
- find/verify clusters
  - is there clear cluster structure in the new low-dim space?
  - does it match a conjectured clustering (color-coded)?
  - ex: glimmer
Example: DR for CG Reflectance Model

- goal: simulate how light bounces off materials to make realistic pictures
  - computer graphics: BRDF (reflectance)
- idea: measure what light does with real materials

[Fig 2. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Capturing Material Reflectance

- measurement: interaction of light with real materials (spheres)
- result: 104 high-res images of material
  - each image 4M pixels

[Fig 5. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Goal: Image Synthesis

- step 1: create new renderings with CG objects that look like captured materials
  - CG teapot looks just like real hematite

- step 2: simulate completely new materials
  - rusty, greasy, ...

[Fig 6, 1. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Need For Low-Dimensional Model

- how to do step 2 simulation of new materials?
  - 104 materials * 4M pixels = 400 million dimensions
  - model much too hi-dim to be useful

- goal: much more concise model that humans can understand/use to generate computer graphics images
  - allow users to tweak meaningful knobs: how shiny, how greasy, how metallic, what color...

- dimensionality reduction to the rescue
Dimensionality Reduction: Linear

- first try: PCA, linear DR technique
- result: error falls off sharply
- good results for step 1 around 45 dims
  - step 2 problem: physically impossible intermediate points when simulating new materials
  - specular highlights cannot have holes!

[Fig 7, 9. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Dimensionality Reduction: Nonlinear

- second try: charting, nonlinear DR
  - better if data embedding is curved not flat

charts = subspace projections of samples (from unknown manifold)
connection = affine merger of charts

[Fig 10. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Dimensionality Reduction: Nonlinear

- second try: charting, nonlinear DR
  - scree plot suggests 10-15 dims
  - note that dim estimate depends on technique used!

[Fig 11. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Finding Semantics for Synthetic Dimensions

- look for meaning in scatterplots
  - each synthetic dimension named by people, not by algorithm
  - points represent real-world images (spheres)
  - people inspect images corresponding to points to decide if axis could have a meaningful name

[Fig 12. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Understanding Synthetic Dimensions

- Crosscheck meaning
- Arrows show simulated images (teapots) made from model
- Check if those match dimension semantics

[Fig 12,16. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Understanding Synthetic Dimensions

Specular-Metallic

[Fig 13,16. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Understanding Synthetic Dimensions

Diffuseness-Glossiness

[Fig 14,16. Matusik et al. A Data-Driven Reflectance Model. SIGGRAPH 2003]
Nonlinear Dimensionality Reduction

- MDS: multidimensional scaling
- confusingly, large family of things all called MDS
  - some linear, some nonlinear!
- classical: minimize strain
  - early formulation equivalent to PCA (linear)
  - spectral methods: approximate eigenvectors
- distance scaling: minimize stress
  - nonlinear optimization
  - force simulation (mass-spring)
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
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
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
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 typical
  - $O(n)$ iteration, $O(n^2)$ algorithm
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

[Fig 1. Ingram, Munzner, and Olano. Glimmer: Multilevel MDS on the GPU. IEEE TVCG, 15(2):249-261, Mar/Apr 2009.]
Glimmer vs Stochastic Alone

- GPU version of stochastic as relaxation subsystem
  - poor convergence properties if run alone
  - only obvious when scalability allows thorough testing

[Fig 2.4. Ingram, Munzner, and Olano. Glimmer: Multilevel MDS on the GPU. IEEE TVCG, 15(2):249-261, Mar/Apr 2009.]
Stochastic Termination

- how do you know when it’s done?
  - no absolute threshold, depends on 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/Verifying Clusters

- sparse document dataset: 28K dims, 28K points
- Glimmer (distance) vs PivotMDS (classical)
  - speed improvement so distance as fast as classical
  - major quality difference for sparse datasets

[Fig 8,9. Ingram, Munzner, and Olano. Glimmer: Multilevel MDS on the GPU. IEEE TVCG, 15(2):249-261, Mar/Apr 2009.]
Showing DR Data

- scatterplot showing points
  - only works if true dimensionality is 2 (... or 3)
  - need to drill down to see what points represent

- SPLOM
  - safe choice

- landscapes
  - avoid! studies show worse than just using points
Hierarchical Parallel Coordinates for Exploration of Large Datasets
Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner,
IEEE Visualization ’99.

Parallel sets: visual analysis of categorical data. Fabien Bendix,

Metric-Based Network Exploration and Multiscale Scatterplot.
Yves Chiricota, Fabien Jourdan, Guy Melancon. Proc. InfoVis 04,
pages 135-142.
Reminders

- Project meetings due 10/19
  - one week from today
- Office hours today after class (5-6)
  - or schedule specific meeting time by email
- No class Oct 24/26