Idiom: boxplot
• static item aggregation
• task: find distribution
• data: table
• derived data
– 5 quant...density. These methods are shown in Figure 5.

Idiom: dynamic filtering
• item filtering
• browse through tightly coupled interaction
– alternative to queries that might return far too many or too few
– pro: straightforward and intuitive
– con: difficult to avoid losing signal
– not mutually exclusive

Idiom: scented widgets
• augment widgets for filtering to show information scent
– cues to 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
• reminder: proposals due by Mon 5pm

Reduce items and attributes
• reduce/increase: inverses
• filter
– pro: straightforward and intuitive
– to understand and compare
– con: out of sight, out of mind
• aggregation
– summary informs about whole set
– con: difficult to avoid losing signal
• not mutually exclusive
– combine filter, aggregate
– combine reduce, change, facet

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 constrained to be smaller than dimensionality of measurements
– issue factors, hidden variables

Evaluating DR
– why do people do DR?
– improve performance of downstream algorithm
– avoid curses of dimensionality
– data analysis
– possible use in output: visual data analysis
– DR tasks
– dimension-oriented task sequences
– node-synthetic dimension, map synthetic dimensions to original axes
– cluster-oriented task sequences
– verify clusters, name clusters, match clusters and classes
**Linear dimensionality reduction**
- principle components analysis (PCA)
  - describe location of each point as linear combination of weights for each axis
  - finding axis: first with most variance, second with next most ...

**Nonlinear dimensionality reduction**
- many techniques proposed
  - MDS, charting, isomap, LLE, TSNE
  - many literatures: visualization, machine learning, optimization, psychology ...
  - can handle curved rather than linear structure
  - cons: lose all ties to original dim/attrs
  - new dimensions cannot be easily related to original

**MDS: Multidimensional Scaling**
- confusingly: entire family of methods, linear and nonlinear!
- classical scaling: minimize strain
  - early formulation equivalent to PCA (linear)
  - Nyström spectral methods approximate eigenvectors: O(N)
  - Landmark MDS (Ts & Xia 2004), PostMDS (Brandes & Pich 2004)
  - limitations: quality for very high dimensional sparse data
- distance scaling: minimize stress
  - nonlinear optimization: O(NP)
  - SPACOF (de Leurue 1977)
  - force-directed placement: O(NP)
  - Stochastic Forces (Chaimers 1994)
  - limitations: quality problems from local minima
- Glimmer goal: O(N^2) 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^4) 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
  - how do you know when it's done?

**Stochastic termination**
- 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)
  - lower 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

**Faster spring model: Stochastic**
- compare distances only with a few points
  - maintain small local neighborhood set
  - O(N) iteration, O(N^2) algorithm
  - small constant: 6 locals, 3 randoms (typically)

**Faster spring model: Stochastic**
- compare distances only with a few points
  - maintain small local neighborhood set
  - stochastic force [Chalmers 1996]

**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

**Methods and outcomes**
- methods
  - quantitative algorithm benchmarks: speed, quality
  - systematic comparison across 8-Q, 10K instances in a few spot checks
  - qualitative judgements of layout quality
- outcomes
  - characterised kinds of datasets where technique yields quality improvements
  - sparse documents
- follow-up work
  - Q1-ONE millions of documents

**Finding and verifying clusters**
- sparse docs dataset
  - 28K dims, 28K points
  - speed equivalent to classical
  - quality major improvement

**Neurocomputing. Special Issue Visual Analytics using Multidimensional Projections, to appear 2014.**

**Glimmer goal: O(N) speed and high quality**