Releviz

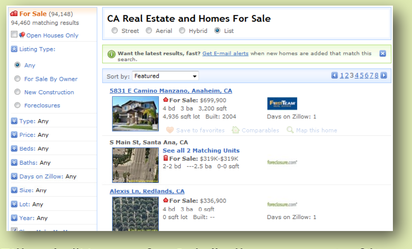
Visualizing similar search results in two dimensions

CS294 Fall 2008: Visualization
David Eitan Poll - depoll@berkeley.edu
Jeff Bowman - jeffbowman@berkeley.edu

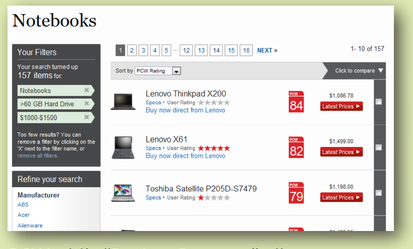
Problem

- Filtering without losing near-misses relies on “relevance”
- Web search engines use an analogous technique for ordering web pages
- Many search tools force users into an iterative process of constraining and relaxing their queries in order to locate useful results
- How can we reduce the number of steps it takes for a user to locate the most useful/ relevant results based upon his query?

Motivation

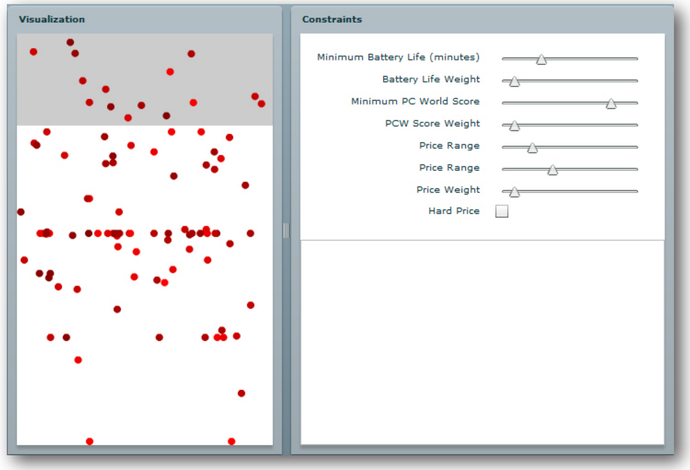


Zillow’s “Homes for Sale” allows users to filter by bedrooms count, lot size, price, and more.



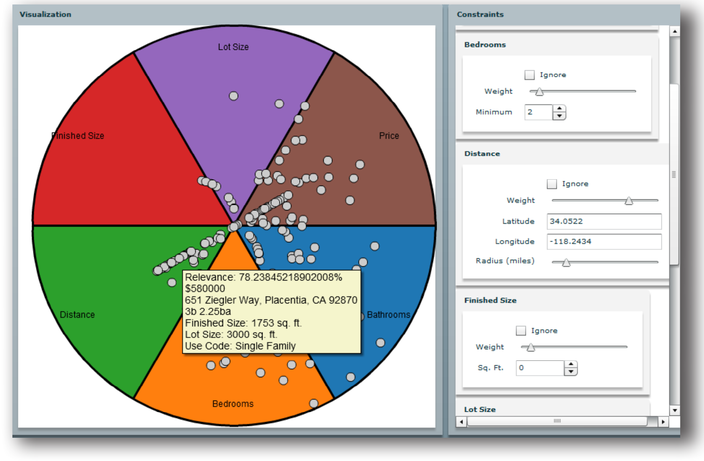
PCWorld’s “Review Browser” allows users to filter laptops by price, manufacturer, and more.

- Today, domain specific search and filtering tools are common
- Many of these tools let users filter results based upon a set of constraints
- What happens when the user over-constrains?
  - Results are not included
  - Users are unaware of “near-misses”
- How can search and filtering engines use visualization to improve the browsing experience?



Linear

- Search engine results model
  - Y coordinate maps relevance, higher = more relevant
  - X coordinate is the grouping
- Selecting a constraint shows what elements are mapped to it
- X-coordinate generated through a non-deterministic force-directed graph
  - Uses a similarity function across all defined constraints
  - The more similar two points are, in terms of their constraint scores, the stronger the pull is between them
  - New X-coordinates calculated offline (points cannot be dragged)
- Animates as the graph changes state



Radial

- Dartboard model
  - Closeness to “bullseye” (radius) maps to relevance.
  - Grouping mapped to angle
- Radius = 1 – relevance
- Θ (angle) generated by:
  - Start with angle mapped to lowest-score constraint
  - Nudge toward each of the remaining violated constraints based upon severity of the violation (low score)
- Animates over polar coordinates
- Background maps constraints to angles and overall scale (perimeter = 0% relevance)

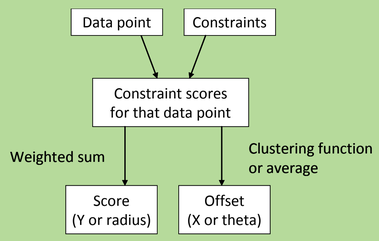
Common Features

- Implemented in Flex 3.0 and Flare (http://flare.prefuse.org)
- Common dataset and constraint object model
  - Constraints can supply their own UI, for added portability
  - Data sets are portable, and define both points and constraints
- Implemented two real-world data sets:
  - Zillow live home sales postings via a web service using Zillow’s SDK/API
  - PCWorld laptop review data stored in a local database
- Interactivity
  - Tooltips
  - Constraint changes cause live updates to visualizations
- Animation, helping users keep context and see the impact of their changes between states and as constraints change

Approach

Constraint satisfaction

- We use constraints to score fitness of a data point
- Constraints are composed of:
  - Scoring function, which takes a data point and returns a value between 0 and 1, and may base its results on user input
  - Weight, which is applied to the score for all data points, used to determine each constraint’s contribution to a data point’s relevance
- Constraints are specific to the data-domain
- The resulting relevance for a data point maps to a particular visual variable



Near Miss Grouping

- If a data point has a lower relevance, users want to know why
- Data points that are unfit for similar reasons can be spatially grouped in order to cue the user as to why this is the case
- For near misses, this generally unambiguous
- The similarity of data points that lose relevance due to multiple constraints is likely to be less meaningful for users
- Hence, we focus on unambiguously representing causes of near-misses.

Future Work

- Explore alternative grouping functions for “similar failure grouping”
- Explore alternative layouts, including 3-dimensional representations
- More encodings allowable for the data points
- Quantitative evaluation through user studies
- Tight integration with other UI Controls, such as hierarchial trees and data grids
- Integration with text search