Chapter 7 - Time-Based Data

Visualizing Change Over Time

Vorlesung „Informationsvisualisierung”
Prof. Dr. Florian Alt, WS 2013/14

Konzept und Folien (4th revised edition):
Thorsten Büring, Andreas Butz, Michael Burch
Outline

• Motivation and Definitions
• Categorisation for Time Dimension
  – Characteristics of the Time Axis
  – Criteria for the „What is analyzed?“ question
  – Criteria for the „How is it represented?“ question
• Examples
  – Historic time-based graphics
  – Timeline representations in Information Visualization
  – Use case: time-based computer desktops
  – Searching time-based data
Examples
LifeStreams

• Freeman & Gelernter 1996
• Use time-ordered stream (and substreams) of documents as organizational metaphor for a computer desktop
• Design objectives
  – No need to name files or to choose a storage location for them
  – Automatic archiving
  – Inherent calendar functionality
  – Personal data should be accessible via a network connection and via any machine
Time Machine

• Hmm, sounds familiar...
• Feature in Mac OS X Leopard, Introduced 2008(?)
TimeScape

- Rekimoto 1999
- Spatial metaphor plus chronological navigation mechanism
  - People tend to pile files on the computer desktop for casual organization (no folders)
  - Time-order system intuitive for archiving
- Items fade out after some time
- Time-travel dial to restore the desktop according to a designated point in time
- Thus users can remove items from the desktop, which currently are not of interest, without actually deleting them
- Users can also travel to the future to make a schedule
- Scheduled objects automatically appear on the desktop at the given time
- Scalable timeline view to support traveling over time
- Keyword search for past and future objects
Landmarks for Timeline Visualization

- Ringel et al. 2003
- Display results of queries on personal content, e.g., web pages, emails, documents
- Overview+detail timeline
  - Summary view showing the distribution of search hits over time
  - Detail view: inspection of individual search results
- Enhance visualization with public landmarks (e.g., holidays, important news events) and personal landmarks (e.g., photos and important calendar events)
- User study indicated significantly faster search times for landmark-augmented timeline compared to a timeline marked only by dates
Landmarks for Timeline Visualization

Overview with hash marks showing the distribution of search results over time

Highlighted region indicates the period displayed in the detail view
Detect Patterns in Time-Series

• Wijk & Selow 1999
• Univariate time-series data
  – Energy consumption over time
  – Number of employees present at work
• How to represent such simple data: draw a graph
• Problems
  – Number of measurements can become very large (e.g., measuring a value every 10 min during a year yields 52,560 values)
  – Repetitive patterns have different scales (seasons, weeks, days)
• Initial attempt
  – Display data as fingerprints
  – Time data is treated as 2D – days and hours are on different axes
Figure 1. Power demand by ECN, displayed as a function of hours and days
Detect Patterns in Time-Series

• Advantages
  – All data is shown simultaneously
  – Seasonal trends can be observed
  – Day pattern can be observed

• Problems
  – Variation over the week is harder to discern
  – Day patterns of Saturdays and Sundays are obscured
  – Smoothing has been applied to make trends more easy to be perceived, but this eliminates fine details

Wijk & Selow 1999
Detect Patterns in Time-Series

• Reduce amount of data to be displayed by using cluster analysis

• Objective: let the analysis tool decide which daily patterns are similar and show their distribution over the year

• Cluster analysis:
  – Split time series into sequence of M day patterns $Y_j$, $j = 1, \ldots, M$
  – Each $Y_j$ consists of seq. of pairs $(y_i, t_i)$, $i = 1, \ldots, N$
  – $y_i$ denotes measured value
  – $t_i$ denotes the time elapsed since midnight
Detect Patterns in Time-Series

- Start with $M$ clusters (each cluster contains one day pattern)
- Compute mutual differences between clusters
- Merge the two most similar clusters to a new cluster
- Repeat clustering until a single large cluster results (contains average of all day patterns)
- Some proposed distance measures ($y_i$ and $z_i$ are values in the day patterns):
  - By average geometric distance
    $$d_{rms} = \sqrt{\frac{\sum (y_i - z_i)^2}{N}}.$$  
  - By similarity of shape (normalize by maximum value in the sequence)
    $$d_{nm} = \sqrt{\frac{\sum (y_i/y_{max} - z_i/z_{max})^2}{N}}.$$  
  - By the difference of peak values
    $$d_{ma} = |y_{max} - z_{max}|.$$  

Eliminate slow trends by subtracting the average difference

$$d_{sh} = \sqrt{\frac{\sum (y_i - z_i - \Delta)^2}{N}},$$  
$$\Delta = \frac{\sum (y_i - z_i)}{N}.$$
Detect Patterns in Time-Series

- Based on cluster analysis a selection of the most significant clusters can be displayed
- Average daily pattern per cluster is shown as graph
- Problem: how to visualize whether similarities of day patterns in a cluster is due to season, week, day etc.?
- Clusters are shown on a calendar, i.e. each day in the calendar is colored according to the cluster to which it belongs

Wijk & Selow 1999
Detect Patterns in Time-Series

Figure 4. Calendar view of the number of employees

Wijk & Selow 1999
Detect Patterns in Time-Series

Figure 5. Cluster analysis of power demand by ECN

Wijk & Selow 1999
TimeSearcher

- Hochheiser & Shneiderman 2004
- Dynamic query tools for searching time series for trends and patterns
- Timebox: rectangular query regions (bounding box) drawn on a 2D representation of time series data
  - X-axis extent: time period of interest
  - Y-axis extent: constraint on the range of values in the time period
- Graphs that do not have values in the given ranges are removed
- Example: 52 weekly stock prices for 1430 stocks
TimeSearcher

- Multiple timeboxes to define conjunctive queries
- Filter hierarchy by order of box creation
- Modifying queries: boxes can be scaled, translated, removed, singly or together
- Box manipulation also via range sliders
- Query by example
  - Pick one graph as query
  - Timeboxes are generated for each data point
  - Boxes are centered around the values of the query graph
  - Manipulate all timeboxes for varying required similarity to graph
TimeSearcher

- Identify leaders & laggards
- Items with behavior trends that anticipate changes of other items in the data set
- Generate new queries by shifting old queries one time period to the right
- Graphs will be displayed that undergo the same transitions similar to leader graphs, i.e. that satisfy the query defined by new timeboxes
- Identify similar trends in other value ranges by translating the boxes
TimeSearcher

• Problem
  – Visual clutter when many graphs are displayed
  – Computational overhead of drawing all graphs and mouse-over handling

• Envelope: contour of extreme values as low-resolution overview
  – Data envelope: all graphs of the data set
  – Query envelope: graphs of the result set

• Threshold to define below which number graphs are displayed
QuerySketch

- Wattenberg 2001
- Query historical stock price data by sketching a graph
- Overview of result set by showing miniature thumbnails + numerical indicator of similarity
- Pattern matching: metric based on the Euclidean distance between sequences of monthly percentage price changes
- Also multiple partial queries possible
- Scale of the graph is logarithmic: stocks with steady growth rate can be queried via a straight line instead of drawing an accurate exponential curve
- Zooming functionality
- Proposed: querying for inequality, e.g. stocks that move above a certain rate
Chapter 8 - Text & Documents

Visualizing and Searching Documents

Vorlesung „Informationsvisualisierung”
Prof. Dr. Florian Alt, WS 2013/14

Konzept und Folien (4th revised edition):
Thorsten Büring, Andreas Butz, Michael Burch
Outline

• Characteristics of text data
• Detecting patterns
  – SeeSoft
  – Arc diagrams
  – Visualizing Plagiarism
  – TextArc
• Keyword search
  – Enhanced scrollbar
  – TileBars
• Cluster Maps
  – Visualization for the document space
  – WEBSOM
  – ThemeScapes
• Cluster map vs. keyword search
Characteristics of Text Data
Text & Documents

• The main vehicle for humans to store information
• Huge existing resources: libraries, WWW
• What to visualize?
• Text is a nominal data type, but with many additional and interesting properties
• Text structure
• Meta data
  – Author
  – Dates
  – Descriptions
• Relations between documents (e.g. citation, similarity)
• Relevance of documents to a query
• Text statistics (e.g., frequency of different words)
• Content / Semantics
Detecting Patterns
PaperLens
SeeSoft

- Eick et al. 1993
- Software visualization tool to display code line statistics (e.g., age, programmer, number of executions in recent test, etc.)
- Encoding
  - Each column represents a file
  - Height of column: length of the document
  - Files exceeding the height of the screen are continued over to the next columns
  - Each row represents a line of code
  - Width of row: length of line
  - Color: age of the line (red: newest; blue: oldest)
- Scales up to 50,000 lines on a single screen
- Example: 20 files with 9,365 lines of code
- Reading windows controlled by virtual magnifying boxes
Arc Diagrams

• Wattenberg 2002
• Visualizes repetition in string data
• Application domains: text, DNA sequences, music
• Approach: to avoid clutter, only visualize an essential subset of all possible pairs of matching substrings
• Display string on a single line
• Connect the consecutive intervals by a semi-circular arc
  – Thickness of the arc: length of the matching substring
  – Height of the arc: proportional to the distance of substrings
Arc Diagrams

- Apply translucency to not obscure matches
- Still: for strings with a high frequency of small repeated substrings the visualization may cause clutter
- Provide users with the ability to filter by minimum substring length to consider
Arc Diagrams

• Comparison to a dotplot diagram
• Recap Matrix diagram
  – Correlation matrix
  – String of n symbols $a_1$, $a_2$, $a_n$ is represented by an n*n matrix
  – Pixel at coordinate (i, j) is black if $a_i = a_j$
  – Can handle very large datasets
  – Shows both small and large-scale structures
• Heavy clutter caused by small substrings with high frequency: n repetitions of a substring lead to $n^2$ visual marks
• Arc Diagrams mark only similar substrings, which are subsequent
Arc Diagrams

• Applied to music, Minuet in G Major, Bach
• Shows classic pattern of a minuet: two main parts, each consisting of a long passage played twice
• Parts are loosely related: bundle of thin arcs connecting the two main parts
• Overlap of the two main arcs shows that the end of the first passage is the same as the beginning of the second passage
Visualizing Plagiarism

• Ribler & Abrams 2000
• Problem: programming assignment in a class with large number of students
• High probability of plagiarism
• Need to compare every document (code file) with every other document
• Visualization must support two steps
  – Highlight suspicious documents
  – Allow for detailed examination of the similar passages - high level of similarity between documents may not be due to cheating (e.g., headers)
Visualizing Plagiarism

- Categorical Patterngram
- Visualize frequencies of sequences of characters present in more than one document
- Remove all non-printable characters in the document collection
- Define length of character sequence to analyse (in the example: 4)
- Histogram-like approach
  - X-axis: start character of sequence
  - Y-axis: number of documents containing the sequence
  - Doc at Y = 1: base document to compare against all other documents
Visualizing Plagiarism

• Composite Categorical Patterngram
• Visualizes which particular documents are similar
• Y-axis: each value corresponds to an individual document
Visualizing Plagiarism

• Case study
• Students were asked to extend a sample program of about 30 lines of code
• Average completed program was about 150 lines
• Submission via email
• Graphic shows categorical patterngram for a single submission
  – Sequence length = 10
  – Lines not text due to high density
  – Rather confusing color coding
• Color coding (not very reasonable)
  – Green: frequency >= 10
  – Red: frequency < 10
  – Blue: base document
• Plagiarism or not?
Visualizing Plagiarism

• What to look out for?
  – Sequences that occur frequently are not of interest - all points with \( y \geq 10 \) are plotted as \( y = 10 \)
  – Suspicious: accumulation of points with low frequencies

• Analysis
  – Majority of points are plotted at \( Y = 1 \)
  – Hence most 10-char sequences are unique to the base document
  – Number of points plotted at \( Y = 2 \), but evenly distributed
Visualizing Plagiarism

- Composite Categorical Patterngram for the submission
- Solid line represents the base document (submission number 23)
- Large number of points plotted in the range of $x = [0; 500]$: email message header
- Other frequent sequences due to the sample program
- Pattern typical for independent work
Visualizing Plagiarism

• Example of patterngrams indicating extensive plagiarism
Visualizing Plagiarism

- Patterngram of more subtle plagiarism
Visualizing Plagiarism

• What may a student do to mask plagiarized code
  – Change variable names
  – Minimize masking effect by replacing all alphanumeric strings in all documents into single characters

• Two documents with the same code but different variable names will produce similar patterngrams
A more recent example [guttenplag.wikia.com]

1218 Plagiatsfragmente aus 135 Quellen auf 371 von 393 Seiten (94.4\%) in 10421 plagiierten Zeilen (63.8\%)

Stand: 03.04.2011 11:55 Uhr

- Seiten, auf denen Plagiate gefunden wurden
- Seiten mit Plagiaten aus mehreren Quellen
- Seiten, auf denen bisher keine Plagiate gefunden wurden
- Das Inhaltsverzeichnis (Seiten 1-14) und die Anhänge (ab Seite 408) wurden nicht bei der Berechnung des Prozentualwertes mit einbezogen
TextArc

- http://www.textarc.org/ - demo
- Represents the entire text as 1 pixel lines in an outer circle
- Text is revealed via mouse-over
- Words are repeated in inner circle at a readable size
- Position of the words depend on where the word appears in the document
- Words that appear throughout the novel will be drawn to the center
- Frequent words stand out
- Example visualizes the novel “Alice in Wonderland”
- Various visualization features
Keyword Search
Search Terms on a Scrollbar

- Byrd 1999
- Searching of keywords in a single document
- Color coding to map each occurrence of a keyword in the document as a small colored icon in the scrollbar
- Provides an overview of the entire document, not only of the portion currently visible
- Users can directly jump to keyword occurrences by moving the slider thumb
TileBars

- Hearst 1995
- Problem with document ranking of common search engines?
- Ranking approach is opaque:
  - What role did the query terms play in the ranking process
  - What is the relationship between the query terms in the document
- TileBars attempts to let the users make informed decisions about which documents and passages to view
TileBars

- Users provide sets of query terms
  - OR within a set
  - AND between sets
- Documents are partitioned into adjacent, non-overlapping multi-paragraph segments
- Each document of the result set is represented by a rectangle - width indicates relative length of the document
- Stacked squares correspond to text segments
- Each row of the stack corresponds to a set of query terms
- Darkness of the square indicates the frequency of terms from the corresponding term set - (Why is this a reasonable color mapping?)
- Title + initial words appear next to each document
- Users can click on segments to retrieve the corresponding text
TileBars

• Analysis hints
  – Overall darkness indicates that all term sets are discussed in detail throughout the document
  – When terms are discussed simultaneously the tiles blend together causing an easy to spot block
  – Scattered term set occurrence show large areas of white space
  – Helps to distinguish between passing remarks and prominent topic terms
• Users may also set distribution constraints to refine the query
  – Minimum number of hits per term set
  – Minimum distribution (percentage of tiles containing at least one hit)
  – Minimum adjacent overlap span