4.6 Temporal Analysis (When)

Science evolves over time. Attribute values of scholarly entities and their diverse aggregations increase and decrease at different rates and respond with different latency rates to internal and external events. Temporal analysis aims to identify the nature of phenomena represented by a sequence of observations such as patterns, trends, seasonality, outliers, and bursts of activity.
A time series is a sequence of events or observations that are ordered in time. Time-series data can be continuous (i.e., there is an observation at every instant of time) or discrete (i.e., observations exist for regularly or irregularly spaced intervals). Temporal aggregations – over journal volumes, years, or decades – are common.
Frequently, some form of filtering is applied to reduce noise and make patterns more salient. Smoothing (i.e., averaging using a smoothing window of a certain width) and curve approximation might be applied. The number of scholarly records is often plotted to get a first idea of the temporal distribution of a dataset. It might be shown in total values or as a percentage of those. One may find out how long a scholarly entity was active; how old it was at a certain point; what growth, latency to peak, or decay rate it has; what correlations with other time series exist; or what trends are observable. Data models such as the least squares model – available in most statistical software packages – are applied to best fit a selected function to a data set and to determine if the trend is significant. Kleinburg's burst detection algorithm is commonly applied to identify words that have experienced a sudden change in frequency of occurrence.

4.6.1 Burst Detection

A scholarly dataset can be understood as a discrete time series, i.e., a sequence of events/observations which are ordered in one dimension – time. Observations (e.g., papers) come into existence at regularly spaced intervals, (e.g., each week, month, issue, volume, or year).
Kleinberg's burst detection algorithm (Kleinberg, 2002) identifies sudden increases in the frequency of words. Rather than using simple frequencies of the occurrences of words, the algorithm employs a probabilistic automaton whose states correspond to increasing frequencies of individual words. State transitions correspond to points in time around which the frequency of the word changes significantly. The algorithm generates a list of the word bursts in the document stream, ranked according to the burst weight, together with the intervals of time in which these bursts occurred. The burst weight depicts the intensity of the burst, i.e., how great the change in the word frequency that triggered the burst. The user must choose if s/he wants to detect bursts related to author names, journal names, country names, references, ISI keywords, or terms used in the title and/or abstract of a paper. This can serve as a means of identifying relevant topics, terms, or concepts that increased in usage, were more active for a period of time, and then faded away.
Kleinberg's burst detection algorithm has four important parameters: the gamma value, first ratio, general ratio, and the number of bursting states. The gamma parameter controls the ease with which the automaton can change states. The higher the gamma value, the smaller the list of bursts generated. The first ratio and general ratio control how great the change of frequency of a word must be to be considered a burst. Usually, only one bursting state is used, since one bursting state is sufficient to indicate whether or not a burst occurred for a specific character string within a specific time period. However, if the user wishes to identify bursts inside bursts, s/he must use more than one bursting state to capture such a hierarchical structure. In Sci2, default values are provided for all parameters.
The Sci2 tool currently uses this algorithm in *'Analysis > Temporal > Burst Detection'*.
Because the algorithm itself is case-sensitive, care must be taken if the user desires 'KOREA' and 'korea' and 'Korea' to be identified as the same word. It is recommended that the user normalize the target data before applying the burst algorithm. The normalization process separates text into word tokens, normalizes word tokens to lower case, removes "s" from the end of words, removes dots from acronyms, deletes stop words, and applies the English Snowball stemmer (http://snowball.tartarus.org/algorithms/english/stemmer.html. The normalization of an entry column can be done using the menu '*Preprocessing > Topical > 'Lowercase, Tokenize, Stem, and Stopword Text'*.
For an example on how to use the burst detection algorithm, see Section 5.2.5 Burst Detection in Physics and Complex Networks (ISI Data).

4.6.2 Slice Table by Time

Slicing a table allows the user to see the evolution of a network over time. Time slices can be cumulative, i.e., later tables include information from all previous intervals, or fully sliced, i.e., each table only includes data from its own time interval. Cumulative slices can be useful for seeing growth over time, whereas fully sliced tables should be used for displaying changing structure over time.

4.6.3 Horizontal Bar Graph

Horizontal bar graphs also allow the user to visualize numeric data over time. This algorithm accepts csv (tabular) datasets, including NSF grant data and the output of burst detections.
In this visualization, the x-axis is time, and the y-axis is amount. The output of this visualization consists of labeled, color-coded horizontal bars that correspond to records in the original dataset.
This algorithm accepts the following input parameters:

- Label: The field used to label the bar lines
- Start Date: The field used for the starting point (on the x-axis) of the records that are visualized as horizontal bars
- End Date: The field used for the ending point (on the x-axis) of the records that are visualized as horizontal bars
- Size Bars By: The field used to size the horizontal bars
- Minimum Amount Per Day For Bar Scaling: The smallest value necessary to allow for scaling the numeric data visualized in the horizontal bars
- Bar Scaling: Can be either linear or logarithmic
- Date Format: The date format to use for parsing the Start Date and End Date.
- Year Label Font Size: The font size to use for year labels. NOTE: If the font size is too large, the labels may overlap.
- Bar Label Font Size: The font size to use for bar labels. NOTE: Extremely large values here may result in poor results.
- Colorized by: The field values used to color code the horizontal bars