Burst Detection

Description

This burst detection algorithm is implemented based on the **Jon Kleinberg's, Bursty and Hierarchical Structure in Streams**. A burst is a period of increased activity, determined by minimizing a cost function that assumes a set of possible states (not bursting and various degrees of burstiness) with increasing event frequencies, where it is expensive (costly) to go up a level and cheap (zero-cost) to decrease a level. It is useful for text stream analysis (such as emails, corpus, publication) where you want to know the activity of the stream in a period of time.

Given a table with at least three columns, a Text Column (event or topics to be targeted), a dates/timestamps (time the event happens) and a delimited value (to separate multiple events / topics), this algorithm detects bursts of each event / topics. Please see 'Usage Hints' for more details about guidance.

The algorithm takes 9 parameters.

- **Gamma** is the value that state transition costs are proportional to. The higher Gamma value results the higher transition costs. Use this parameter to control how ease the automaton can change states.
- **Density scaling** determines how much 'more bursty' each level is beyond the previous one. The higher the scaling value, the more active (bursty) the event happens in each level.
- **Bursting States** determines how many bursting states there will be, beyond the non-bursting state. An i value of bursting states is equals to i+1 automaton states.
- **Date Column** is the name of the column with date/time when the events / topics happens.
- **Date Format** specifies how the date column will be interpreted as a date/time. See http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html for details.
- **Burst Length Unit** specifies how to divide the date range into burstable units.
- **Burst Length** specifies the number of burstable units per burstable period. For example, 10 years generates bursts by decade.
- **Text Column** is the name of the column with values (delimiter and tokens) to be computed for bursting results.
- **Text Separator** delimits the tokens in the text column. When constructing your tables, do not use a separator that is used as a whole or part of any token.

The result will be generated into a CSV file with the following fields:

- **Word** is the target event / topic
- **Level** is the burst level of this burst. The higher burst level, the more frequent the event / topic happens.
- **Weight** is the weight of this burst between its **Length**. A higher weight could be resulted by the longer **Length**, the higher **Level** or both.
- **Length** is the period of the burst. It is generated based on (**Start** - **End** + 1).
- **Start** is the starting time of this burst
- **End** is the ending time of this burst

Pros & Cons

Because of the by-event state machine approach, events are bursted on independently of each other. This makes this algorithm suitable primarily when the changes in patterns of individual event usage are the area of interest. Cross-burst-levels comparisons of an event are possibly using the burst 'weight'. However, this algorithm only support batches records by years, months, days, hours, minutes.

Applications

Burst detection is particularly useful for examining the trends in collections of texts or communities of conversation. Even words that are used comparatively little, but that change in frequency of usage over time, stand out, unlike in burst detection algorithms based on thresholds.

Implementation Details

Since we are focus on scholarly data, the data will be distributed into batches (usually yearly batches) before the burst computation started. The burst detection algorithm was re-implemented in Java based on the origin C implementation. Please see Kleinberg [pg. 14]. We replace the missing years with empty batches to make the batches continuously by year. There will no burst for these empty batches. Users can change the scaling factor for the batches to month, day, hour; even number of years per batch. The batching implementation will not consider the date fields with the scaling factors that are smaller than the user selected scaling factor. For example, if the days scaling factor is selected, the batching algorithm will remove the hour and minute fields in the date value.

Usage Hints

Please read the Description section before continuing. This burst algorithm is a text based burst detection that provide burst results in hierarchical structure. However, it is also capable to detect if the bursts exist by setting the bursting states to 1.

Preprocessing steps:
You might need to consider to normalize free-form text of the **Text Column** by using Lowercase, Tokenize, Stem, and Stopword Text. The burst detection algorithm will not edit the words in the **Text Column**. The different forms of a word such as author, Author, authors will be treated as different tokens. To avoid this, you can use the Lowercase, Tokenize, Stem, and Stopword Text algorithm to normalized the **Text Column**. Basically, the normalized result is a list of tokens (words) that delimited with '|'.

Steps:

1. Load the CSV file by choosing load from menu bar.
2. Choose Analysis > Topical > Burst Detection from the Sci2's menu bar.
3. A window will popup and a 7 input parameters are listed.
4. Usually you will only need to change Date Column, Date Format, Burst Length Unit, Burst Length, Text Column, and Text Separator when using Burst Detection. If you want to have a abstract view of the entire data, set the bursting states to 1. If you interested to the hierarchical burst structure of each word, set the bursting states bigger than 1. Change the parameters based on your need. Please see 'Desciption' section for detail of each parameter.

5. Press ok once you done adjusting parameters. A new result in csv file will be generated at the right panel. Save it and you can view the result with Excel.
6. Please refer to description for result fields information.

Visualization steps:

1. Please use the Horizontal Bar Graph to visualize the results of bursting states equals to 1. The visualization supports of the hierarchical result is still under development.

Notes: The default parameters are typically good choices, but more sophisticated models can be fitted by tweaking them in various ways. You might not want to include the records with empty Text Column into the input file if you don't want to count them into the burst result. This could be happens while you have a set of records from year 1970 - 2011. But there are no text information between 1970 - 1980. Since it is lack of information rather than no burst, you might want to remove records in year 1970 - 1980 to have a better focus period.

Links

- Source Code
- Home Page: http://iv.slis.indiana.edu/sw/burst.html (describes the C implementation)

References

J. Kleinberg. Bursty and Hierarchical Structure in Streams. Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2002.

See Also