Stream Clustering of Chat Messages with Applications to Twitch Streams

Matthias Carnein, Dennis Assenmacher, Heike Trautmann
Group for Information Systems and Statistics
Data Mining

- Data mining is core to many business models today
  - Google, Facebook, Insurance companies, Fashion retailers, ...
- It involves analysing enormous data and extracting useful information from it

Data of Today:
Fast and Unstructured
Clustering

- Clustering is an important tool in data mining
- Discover unknown patterns, i.e. groups of similar objects
Objective

Design a clustering algorithm for streams of text data which . . .

1. allows to identify topics, e.g. in chats or social media
2. is efficient and can handle very fast streams
3. is applicable to texts of arbitrary length, language and content

Application scenarios include social media analysis and customer service
Outline

1. Background
2. Stream Clustering of Text
3. Evaluation
4. Conclusion
Outline

1. Background

2. Stream Clustering of Text

3. Evaluation

4. Conclusion
Stream Clustering

- Continuous stream of observations
- The order of data points cannot be influenced and not all points can be revisited or stored

Maintain a set of valid clusters
Two-Stage Clustering

Data stream

Online

'Micro-clusters'

Offline

'Reclustering'

'Macro-clusters'

[Aggarwal et al. 2003]
Features of Texts

Sentence

unigrams

bigrams

trigrams

This is a sentence

This is is a a sentence

This is is a a sentence

This is is a a sentence
TF-IDF

- **Term Frequency (TF)**

  Message: How do you do

<table>
<thead>
<tr>
<th>How</th>
<th>do</th>
<th>you</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Inverted Document Frequency (IDF)**
  - Reduce influence of words that are common in all documents

  Message 1: I like you
  Message 2: I hate you

<table>
<thead>
<tr>
<th>I</th>
<th>you</th>
<th>hate</th>
<th>like</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>you</td>
<td>hate</td>
<td>like</td>
</tr>
</tbody>
</table>

  more important
Cosine Similarity

- **Similarity:** cosine of the angle between two TF-IDF vectors

\[
\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}
\]

\[
\theta
\]

\[
\text{I}
\]

\[
\text{like}
\]

\[
\text{hate}
\]

\[
\text{I like}
\]

\[
\text{I hate}
\]
Outline

1. Background

2. Stream Clustering of Text

3. Evaluation

4. Conclusion
Stream Clustering of Text

Based on a distance threshold merge into existing cluster or initialize a new one.
Damped Time Window

- Assign a weight to all clusters
- Increment on update and decay in every time step
- Remove cluster which decay below a threshold

\[ f(\Delta t) = 2^{-\lambda \Delta t} \]
Damped Time Window

- Assign a weight to all clusters
- Increment on update and decay in every time step
- Remove cluster which decay below a threshold

<table>
<thead>
<tr>
<th>Word</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>haha</td>
<td>5.0</td>
</tr>
<tr>
<td>funny</td>
<td>1.4</td>
</tr>
<tr>
<td>job</td>
<td>3.1</td>
</tr>
<tr>
<td>nice</td>
<td>5.0</td>
</tr>
<tr>
<td>url</td>
<td>1.4</td>
</tr>
<tr>
<td>link</td>
<td>3.1</td>
</tr>
<tr>
<td>please</td>
<td>5.0</td>
</tr>
<tr>
<td>weight:</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Damped Time Window

- Assign a weight to all clusters
- Increment on update and decay in every time step
- Remove cluster which decay below a threshold

Optional:
- Also decay term frequencies
- Reduces memory requirements
Periodic Cleanup

To improve the clustering, we perform periodic adjustments

Fading

- Fading and evaluating all weights can be costly
- Only update weights periodically

Merging

- Merge clusters that moved in close proximity
- Such clusters likely represent the same topic
Reclustering

- The result of our algorithm is a number of small discussion threads
- Recluster into overall topics using any distance based clustering algorithm, e.g. hierarchical clustering
Outline

1. Background

2. Stream Clustering of Text

3. Evaluation

4. Conclusion
Twitch

- Streaming video platform with a focus on video games
- 1.5 million broadcasters and 100 million visitors per month
- Key component: Interactive chat based on IRC
  → Allows real time monitoring

Sable: I used to see these numbers 30k plus
Jahmonaut: csgo in general has gone downhill
Sable: Yeah, hoping the dust update brings it back
Jahmonaut: heck yee
tendria: VAC = Valve Allow Cheats
Sable: Yeah, Speaking cheats, I ran into a lovely spinofter last night
Sable: wasn't fun
Jahmonaut: V - bringA - backC - Dust
callesoe: Hello
Houragorn: 30fps really
12pcrs: 700 viewers on major
Kamyab: yeah csgo is going downhill
Moobot: Join the ESLCS steam group
http://steamcommunity.com/groups/eslcs
k3n3sto: why does the m4 sound like a galil?
Like YouTube, but live!
Data Collection

- Selected the 10 most popular channels
- Collected all messages for 3 days
- More than 1.1 Million chat messages
Data Collection

- Treat collected data as a single continuous stream
- Use channel as the true class label
- Check if clusters reveal the true channels
Prequential evaluation [Cao et al. 2006]
Parameter Tuning

- Choosing the best parameters is difficult and time consuming

  → Automatic parameter-configuration using Iterated Racing (irace) [López-Ibáñez et al. 2016]

Settings

- Search for configurations that optimize cluster purity
- Upper limit of 500 micro-clusters
Results

- **Purity**: Proportion of points that belong to the majority class in a cluster

![Graph showing Purity vs Message with and without token fading]
Results

- **Silhouette Width**: how similar an observation is to its own cluster, compared to other clusters

![Silhouette Width Comparison](image)
Processing Speed

- With token fading
- Without token fading

![Graph showing processing speed with and without token fading.](image-url)
Actual vs. Processed Speed

Consistently faster processing despite combining 10 streams
Outline

1. Background

2. Stream Clustering of Text

3. Evaluation

4. Conclusion
Conclusion

▶ New clustering algorithm for text streams
▶ Application scenarios include social media analysis and customer service
▶ The algorithm allows to identify topics, e.g. in chats
▶ Applicable to texts of arbitrary length, language and content
▶ We demonstrated our approach on more than 1 Million chat messages from Twitch

Outlook

▶ Evaluate on longer data, e.g. emails
▶ Adaptively choose parameters
Questions?
References 1

Aggarwal, Charu C., Jiawei Han, Jianyong Wang and Philip S. Yu (2003). ‘A Framework for Clustering Evolving Data Streams’. In: Proceedings of the 29th International Conference on Very Large Data Bases. Vol. 29. VLDB ’03. Berlin, Germany: VLDB Endowment, pp. 81–92.
