INF 553 Mini Project

Siqi Liang
3330505568

Nov. 2019

TopicSketch: Real-Time Bursty Topic Detection from Twitter[1]

Model & Solutions

In daily social media like Twitter, special events like "NBA final game", "earthquake" will trigger a large amount of relevant tweets in a short time, which we call "bursty topics". *TopicSketch* is a framework for detection of such event in real-time.

First, it defines what could be considered as "bursty topics". In a nutshell, this paper uses frequency of word pairs and triples to calculate the velocity and acceleration of words, and only words related to bursty topics will have significant larger acceleration compared with non-bursty topic words, which is an intuitive idea. And the "velocity" $\hat{v}(t)$ and "acceleration" $\hat{a}(t)$ are defined as

$$\hat{v}_{\Delta T}(t) = \sum_{t_i<t} X_i \cdot \frac{\exp((t_i - t)/\Delta T)}{\Delta T}$$

$$\hat{a}(t) = \frac{\hat{v}_{\Delta T_2}(t) - \hat{v}_{\Delta T_1}(t)}{\Delta T_1 - \Delta T_2},$$

where X_i is the frequency of the words.

Then, considering single topic model, assume there are K topics $\{\phi_k\}_{k=1}^K$, where $\phi_k \in [0,1]^W$ is a distribution over words, where W is the number of distinct words in the whole documents. For each tweet d_i, there is only one possible topic z_i, which is unobservable. And each word in d_i is drawn from $Multinomial(\phi_{z_i})$. Once the topics $\{\phi_k\}_{k=1}^K$ and topic indicator z_i are fixed for d_i, the paper asserts that

$$\mathbb{E}[f_{1,i}[w]] = \phi_{z_i}[w],$$

where $f_{1,i}[w] = \frac{C_{i,w}}{C_i}$ is the frequency of a word w in d_i. And the frequency of a word pair (w_1, w_2) in a tweet d_i with C_i words can be denoted as follows,

$$f_{2,i}[w_1, w_2] = \begin{cases} \frac{P(C_{i,w_1},2)}{P(C_i,2)} & w_1 = w_2 \\ \frac{C_{i,w_1} C_{i,w_2}}{P(C_i,2)} & w_1 \neq w_2. \end{cases}$$

And it can be proven

$$\mathbb{E}[f_{2,i}[w_1, w_2]] = \phi_{z_i}[w_1] \cdot \phi_{z_i}[w_2].$$

For all possible (w_1, w_2), the matrix $f_{2,i}$ can be expressed as

$$\mathbb{E}[f_{2,i}] = \phi_{z_i} \otimes \phi_{z_i}.$$

And similarly, the frequency of a word triple (w_1, w_2, w_3) in a tweet d_i will have

$$\mathbb{E}[f_{3,i}] = \phi_{z_i} \otimes \phi_{z_i} \otimes \phi_{z_i}.$$

Then based on truth that acceleration $\mathcal{A}_t(\{X_i\})$ can be expressed as linear combination of $\{X_i\}$, it's easy to have

$$\mathbb{E}[\mathcal{A}_t(\{f_{2,i}\})] = \sum_k \mathcal{A}_t(\{\mathbf{1}_k(z_i)\}) \cdot \phi_k \otimes \phi_k,$$
$$\mathbb{E}[\mathcal{A}_t(\{f_{3,i}\})] = \sum_k \mathcal{A}_t(\{\mathbf{1}_k(z_i)\}) \cdot \phi_k \otimes \phi_k \otimes \phi_k.$$

Now the problem becomes, given tweets stream $D = \{d_i\}$,

1) obtain frequency matrix $M_2 = \mathcal{A}_t(\{f_{2,i}\}) \in \mathbb{R}^{N \times N}$ and frequency tensor $M_3 = \mathcal{A}_t(\{f_{3,i}\}) \in \mathbb{R}^{N \times N}$ from $D = \{d_i\}$, which are also called *sketch data*;

2) find $\{\phi_k\}_{k=1}^K$ using M_2, M_3;

3) Find active words for each topic in $\{\phi_k\}_{k=1}^K$.

Step 1) is implemented using lazy strategy: efficiently compute acceleration by incrementally maintain two velocities $\hat{v}_{\Delta T_1}(t)$ and $\hat{v}_{\Delta T_2}(t)$; only update the pair $(\hat{v}_{\Delta T_1}(t),\ \hat{v}_{\Delta T_2}(t))$ only if $C_{i,w_1} \cdot C_{i,w_2} > 0$. Also, the framework updates sketch matrices $M_2 = \mathcal{A}_t(\{f_{2,i}\}_{d_i \in D})$ and $M_3 = \mathcal{A}_t(\{f_{3,i}\}_{d_i \in D})$ in parallel, that is, update partial matrix on different processes/cores at the same time, then obtain the final matrix by adding them up.
After obtaining sketch data, the paper use Min-Count hash method to reduce the matrix/tensor dimensions using H hash functions, which results in H small sketch matrices $\{M_2^{(h)}\}_{h=1}^H$ and $\{M_3^{(h)}\}_{h=1}^H$, with each matrix $M_2^{(h)}$, $M_3^{(h)}$ of shape $B \times B$ (reduce matrix from $N \times N$ to $B \times B$).
For step 2), the paper propose a tensor decomposition algorithm to find $\{\phi_k\}_{k=1}^K$, in which SVD is used. And in practical implementation, they decompose each $M_2^{(h)}$ and $M_3^{(h)}$ in parallel to speed up the computation, which results in $\{\phi_k^{(h)}\}_{h=1}^H$ for each $k \in [K]$. Now the computation complexity is $O(H \cdot B^2 \cdot K)$.
For step 3), the paper proposes *TopicRecover* algorithm to find active words for each $topic_k$ using $\{\phi_k^{(h)}\}_{h=1}^H$, that is, for each word w in candidate active word pool, if $\min_{1 \le h \le H}\{\phi_k^{(h)}[\mathcal{H}_h(w)]\} \ge threshold$, then add word w to $topic_k$. Finally, this step will output active word list for each $topic_k$.
In practice, the framework adds heuristic step to avoid the problem caused by hashing collision and spam accounts:

i) Trivial topic filtering

ii) Noisy topic filtering

iii) Spam filtering

iv) Rare word pruning

Highlights

i) Use both frequency of word pair as well as word triples, rather than only word pairs.

ii) Use dimension reduction to reduce computation complexity while remain information.

iii) Parallelized framework which could detect topics in real-time.

Limitation

i) Fragile when facing spam accounts (refined using spam filtering)

ii) Not suitable for stream of documents with multiple topics

Related Works

The original paper focused on other models on bursty topic detection in this section, however here we focus on NLP models using tensor methods.
Cohen and Collins [2] proposed inference with latent-variable PCFGs based on tensor formulation and used tensor decomposition to speed up the parsing process. Chang et al. [3] proposed tensor decomposition algorithm for knowledge base embedding to extract relation from entity-relation triples. Wang et al. [4] proposed fast CP tensor decomposition algorithm combined with hashing functions for dimension reduction, and applied this on topic modeling.

References

[1] Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. Topicsketch: Real-time bursty topic detection from twitter. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2216–2229, 2016.

[2] Michael Collins and Shay B Cohen. Tensor decomposition for fast parsing with latent-variable pcfgs. In *Advances in Neural Information Processing Systems*, pages 2519–2527, 2012.

[3] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, 2014.

[4] Yining Wang, Hsiao-Yu Tung, Alexander J Smola, and Anima Anandkumar. Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems*, pages 991–999, 2015.