Real-Time Top-R Topic Detection on Twitter with Topic Hijack Filtering

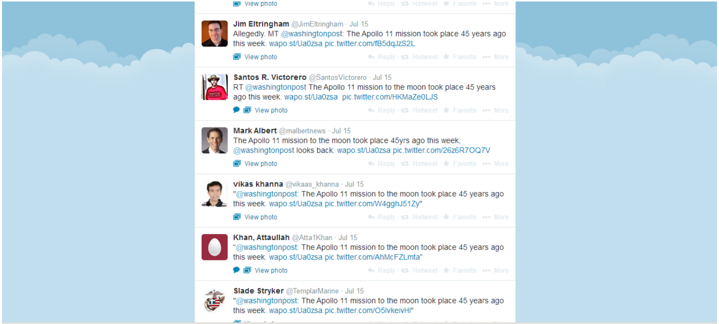
Kohei Hayashi (NII & ERATO)

August 3, 2015

Joint work with

- Takanori Maehara (Shizuoka Univ)
- Masashi Toyoda (Univ Tokyo)
- Ken-ichi Kawarabayashi (NII & ERATO)

Twitter



SNS with short messages (tweets)

- **Big data** 41M users, 1.4B interactions
- **Diversity** Covering any topics: news, politics, TV, ...
- **Rapid** 1 tweet $\leq$ 140 chars
  $\Rightarrow$ Low latency

Automatic Trend Detection on Twitter

Automatic Trend Detection on Twitter

A promising data resource for topic detection

- Find word clusters by *word co-occurrence*
- May discover breaking news and events even faster than news media

Two Challenges

- Topic Detection in Real-time
- Noise Filtering

Topic Detection in Real-time

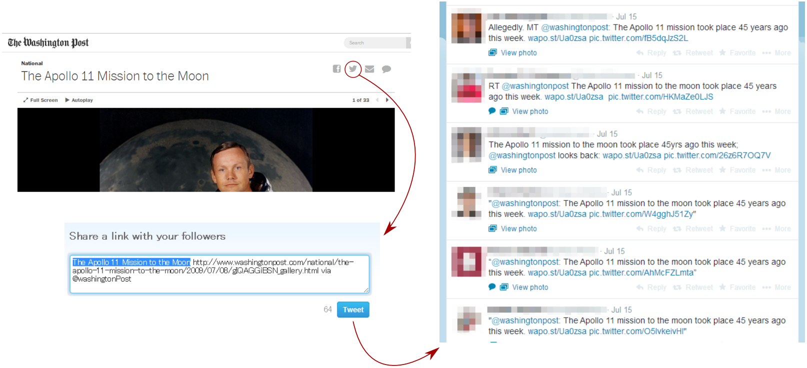
An ultimate goal of topic detection on Twitter

- Have to deal with 0.27M tweets/min
- Words rarely co-occur
  $\Rightarrow$ Severely degrade the quality of topics

Noise Filtering

Many spam tweets generated by not human

- e.g. "tweet buttons"



Exaggerate co-occurrence and "hijack" important topics

Contributions

A streaming topic detection algorithm based on *non-negative matrix factorization (NMF)*

1. Highly scalable:
   Able to deal with a 20M$\times$1M sparse matrix/*sec*
2. Automatic topic hijacking detection & elimination

Contributions

A streaming topic detection algorithm based on *non-negative matrix factorization (NMF)*

1. Highly scalable:
   Able to deal with a 20M$\times$1M sparse matrix/*sec*
2. Automatic topic hijacking detection & elimination

Technical Points

1. Reformulate NMF in a stochastic manner
   - Stochastic gradient descent (SGD) updates with $O(\mathrm{NNZ})$ time
2. Use of statistical testing
   - Assume normal topics follow *power law*

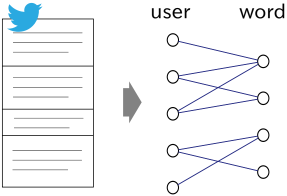
Streaming NMF

Topic Detection by NMF

Consider to obtain R topics from all past tweets

Tweets

User-word co-occurrence

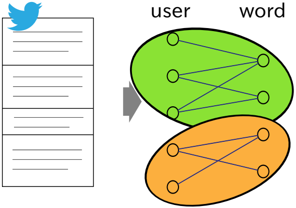


Topic Detection by NMF

Consider to obtain R topics from all past tweets

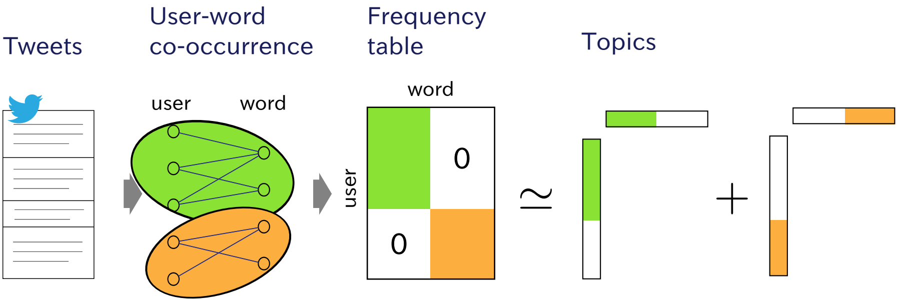
Tweets

User-word co-occurrence



Topic Detection by NMF

Consider to obtain R topics from all past tweets



$$\mathbf{X} \simeq \sum_{r=1}^{R} \mathbf{u}_r \mathbf{v}_r^\top$$

Problem

$$\min_{\mathbf{U}\in\mathbb{R}_+^{I\times R},\mathbf{V}\in\mathbb{R}_+^{J\times R}} f_\lambda(\mathbf{X};\mathbf{U},\mathbf{V}),$$

$$f_\lambda(\mathbf{X};\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{X}-\mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 + \frac{\lambda}{2}(\|\mathbf{U}\|_{\mathrm{F}}^2 + \|\mathbf{V}\|_{\mathrm{F}}^2)$$

Batch Algorithm

Repeat until convergence:

1. $\mathbf{U} \leftarrow [\mathbf{U} - \eta\nabla_{\mathbf{U}} f_\lambda(\mathbf{X};\mathbf{U},\mathbf{V})]_+$
2. $\mathbf{V} \leftarrow [\mathbf{V} - \eta\nabla_{\mathbf{V}} f_\lambda(\mathbf{X};\mathbf{U},\mathbf{V})]_+$

Guaranteed converging to stationary points

Stochastic Formulation

- Now we observe $\mathbf{X}^{(t)}$ for each time t
- Keep track to $\bar{\mathbf{X}}^{(t)} = \frac{1}{t}\sum_{s=1}^{t} \mathbf{X}^{(s)}$
  - Efficient updates from $\bar{\mathbf{X}}^{(t)}$ to $\bar{\mathbf{X}}^{(t+1)}$?

Stochastic Formulation

- Now we observe $\mathbf{X}^{(t)}$ for each time t
- Keep track to $\bar{\mathbf{X}}^{(t)} = \frac{1}{t}\sum_{s=1}^{t} \mathbf{X}^{(s)}$
  - Efficient updates from $\bar{\mathbf{X}}^{(t)}$ to $\bar{\mathbf{X}}^{(t+1)}$?

Key idea: decompose f_λ for each t

$$\|\bar{\mathbf{X}}^{(t)} - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 = \frac{1}{t}\sum_{s} \|\mathbf{X}^{(s)} - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 + \mathrm{const.}$$

Stochastic Formulation

- Now we observe $\mathbf{X}^{(t)}$ for each time t
- Keep track to $\bar{\mathbf{X}}^{(t)} = \frac{1}{t}\sum_{s=1}^{t} \mathbf{X}^{(s)}$
  - Efficient updates from $\bar{\mathbf{X}}^{(t)}$ to $\bar{\mathbf{X}}^{(t+1)}$?

Key idea: decompose f_λ for each t

$$\|\bar{\mathbf{X}}^{(t)} - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 = \frac{1}{t}\sum_{s} \|\mathbf{X}^{(s)} - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 + \mathrm{const.}$$

If $\mathbf{X}^{(t)}$ is i.i.d. random variable,

$$\|\mathbb{E}[\mathbf{X}^{(t)}] - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 = \mathbb{E}\|\mathbf{X}^{(t)} - \mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2 + \mathrm{var}[\mathbf{X}^{(t)}]$$

Now we can use stochastic optimization

Streaming Algorithm

For $t = 1, \ldots, T$:

1. $\mathbf{A}_U \leftarrow \nabla_{\mathbf{U}} \nabla_{\mathbf{U}} f_\lambda, \quad \mathbf{A}_V \leftarrow \nabla_{\mathbf{V}} \nabla_{\mathbf{V}} f_\lambda$ (metrics)
2. $\mathbf{U}^{(t)} \leftarrow [\mathbf{U}^{(t-1)} - \eta_t \nabla_{\mathbf{U}} f_\lambda(\mathbf{X}^{(t)}; \mathbf{U}, \mathbf{V}^{(t-1)}) \mathbf{A}_U^{-1}]_+$
3. $\mathbf{V}^{(t)} \leftarrow [\mathbf{V}^{(t-1)} - \eta_t \nabla_{\mathbf{V}} f_\lambda(\mathbf{X}^{(t)}; \mathbf{U}^{(t)}, \mathbf{V}) \mathbf{A}_V^{-1}]_+$

Guaranteed converging to the stationary points of $f_\lambda(\bar{\mathbf{X}}^{(t)}; \mathbf{U}, \mathbf{V})$ for some η_n and i.i.d. $\{\mathbf{X}^{(t)}\}$

Comparing with the Batch NMF ...

Much faster

- Update cost: depends on $\mathsf{NNZ}(\mathbf{X}^{(t)}) \ll \mathsf{NNZ}(\bar{\mathbf{X}}^{(t)})$

Memory efficient

- Able to discard $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(t-1)}$

Comparing with the Batch NMF ...

Much faster

- Update cost: depends on NNZ($\mathbf{X}^{(t)}$) $\ll$ NNZ($\bar{\mathbf{X}}^{(t)}$)

Memory efficient

- Able to discard $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(t-1)}$

Smoothing effect

$$
\begin{aligned}
\mathbf{U}^{(t)} &= [(1-\eta_t)\mathbf{U}^{(t-1)} + \eta_t \mathbf{X}^{(t)} \mathbf{V}^{(t-1)} \mathbf{A}_{\mathbf{U}}^{-1}]_+ \\
&= [(1-\eta_t)\mathbf{U}^{(t-1)} + \eta_t \operatorname*{argmin}_{\mathbf{U}} f_\lambda(\mathbf{X}^{(t)}; \mathbf{U}, \mathbf{V}^{(t-1)})]_+
\end{aligned}
$$

- A weighted average of
  the prev solution and the NMF solution of $\mathbf{X}^{(t)}$
- Mitigates the sparsity of $\mathbf{X}^{(t)}$

Topic Hijacking Detection

Problem Setting

Goal: Find hijacked topics

Idea: Check word distributions

- The word dist of *a hijacked topic* should be different from the word dist of *a normal topic*

Problem Setting

Goal: Find hijacked topics

Idea: Check word distributions

- The word dist of *a hijacked topic* should be different from the word dist of *a normal topic*
- NMF estimates topic-specific word dists as $\mathbf{V}$

$$
\begin{aligned}
X_{ij} &\propto p(\mathrm{user}_i, \mathrm{word}_j) \\
&\propto \sum_r p(\mathrm{topic}_r) p(\mathrm{user}_i | \mathrm{topic}_r) p(\mathrm{word}_j | \mathrm{topic}_r) \\
&\propto \sum_r u_{ir} v_{jr}
\end{aligned}
$$

Defining Normal/Hijacked Topics

Normal topics:

- Many users involve
  $\Rightarrow$ Mixing many different vocabularies
  $\Rightarrow$ Heavy-tailed (Zipf's law)

Defining Normal/Hijacked Topics

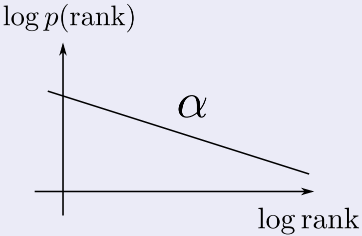
Normal topics:

- Many users involve
  $\Rightarrow$ Mixing many different vocabularies
  $\Rightarrow$ Heavy-tailed (Zipf's law)

Definition: A Normal Topic

Topic r is normal if

$$p(\mathrm{rank}(\mathrm{word}) \mid \mathrm{topic}_r) = \mathrm{power}(\alpha)$$



Defining Normal/Hijacked Topics (Cont'd)

Hijacked topics:

- Few users involve
  $\Rightarrow$ The same vocabulary is repeatedly used
  $\Rightarrow$ Uniform probs & (almost) no tail

Defining Normal/Hijacked Topics (Cont'd)

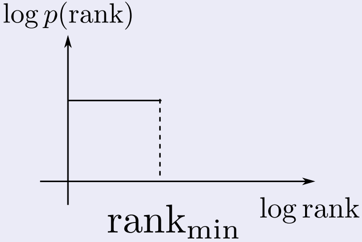
Hijacked topics:

- Few users involve
  $\Rightarrow$ The same vocabulary is repeatedly used
  $\Rightarrow$ Uniform probs & (almost) no tail

Definition: A Hijacked Topic

Topic r is hijacked if

$$p(\mathrm{rank}(\mathrm{word}) \mid \mathrm{topic}_r) = \mathrm{step}(\mathrm{rank}_{\mathrm{min}})$$



Log-likelihood Ratio Test

$$\mathcal{L}(\mathrm{rank}_{\min}) = \sum_j \log \frac{\mathrm{step}(\mathrm{rank}_j \mid \mathrm{rank}_{\min})}{\mathrm{power}(\mathrm{rank}_j \mid \hat{\alpha})}$$

Theorem (Asymptotic normality [Vuong'89])

Let N be # of observed words. Then, $\mathcal{L}(\mathrm{rank}_{\min})/\sqrt{N}$ converges in distribution to $N(0, \sigma^2)$ where

$$\sigma^2 = \frac{1}{N}\sum_{j=1}^{N}\left(\log \frac{\mathrm{step}(\mathrm{rank}_j \mid \mathrm{rank}_{\min})}{\mathrm{power}(\mathrm{rank}_j \mid \hat{\alpha})}\right)^2 - \left(\frac{1}{N}\mathcal{L}(\mathrm{rank}_{\min})\right)^2.$$

Log-likelihood Ratio Test

$$\mathcal{L}(\mathrm{rank}_{\min}) = \sum_j \log \frac{\mathrm{step}(\mathrm{rank}_j \mid \mathrm{rank}_{\min})}{\mathrm{power}(\mathrm{rank}_j \mid \hat{\alpha})}$$

Theorem (Asymptotic normality [Vuong'89])

Let N be # of observed words. Then, $\mathcal{L}(\mathrm{rank}_{\min})/\sqrt{N}$ converges in distribution to $N(0, \sigma^2)$ where

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{N} \left(\log \frac{\mathrm{step}(\mathrm{rank}_j \mid \mathrm{rank}_{\min})}{\mathrm{power}(\mathrm{rank}_j \mid \hat{\alpha})} \right)^2 - \left(\frac{1}{N} \mathcal{L}(\mathrm{rank}_{\min}) \right)^2 .$$

For $r = 1, \ldots, R$:

- Estimate $\hat{\alpha} = \mathrm{argmax}_{\alpha} \log \mathrm{power}(\mathrm{rank}(\mathbf{u}_k) \mid \alpha)$
- For $\mathrm{rank}_{\min} = 1, \ldots, 140$:
  - Compute $\mathcal{L}(\mathrm{rank}_{\min})$
  - Topic r is hijacked if p-val < 0.05

Experiments

Data

Japanese Twitter stream

- April 15–16, 2013
  - 417K users
  - 1.98M words
  - 15.3M tweets
  - 69.4M co-occurrences
- Generated $\mathbf{X}^{(t)}$ for each 10K co-occurrences

Runtime

	NMF	1%	10%	100%
Batch		27.0m	1.9h	16.4h
Online	[Cao+ 07]	5.6h	9.2h	17.8h
Dynamic	[Saha+ 12]	16.7h	>24h	>24h
Streaming	[proposed]	4.0m	21.7m	3.6h
w/ Filter	[proposed]	5.3m	24.1m	3.8h

Runtime

	NMF	1%	10%	100%
Batch		27.0m	1.9h	16.4h
Online	[Cao+ 07]	5.6h	9.2h	17.8h
Dynamic	[Saha+ 12]	16.7h	>24h	>24h
Streaming	[proposed]	4.0m	21.7m	3.6h
w/ Filter	[proposed]	5.3m	24.1m	3.8h

✔ Streaming NMF: $\times$5–250 faster!

- 67K tweets/m

  $\Rightarrow$ The real-time speed of all jp tweets!

✔ Filtering cost is ignorable

Perplexity

- Similarity between a topic dist and a target dist
  - Used Y! headlines[1] for the target term dist
- Lower is better

	NMF	1%	10%	100%
Batch		9.01E+09	6.32E+06	4.51E+04
Online	[Cao+ 07]	1.06E+04	3.25E+04	1.83E+04
Dynamic	[Saha+ 12]	6.53E+04	N/A	N/A
Streaming	[proposed]	5.65E+07	3.25E+04	8.71E+03
w/ Filter	[proposed]	5.25E+09	2.40E+04	7.90E+03

[1] `http://news.yahoo.co.jp/list`

Perplexity

- Similarity between a topic dist and a target dist
  - Used Y! headlines[1] for the target term dist
- Lower is better

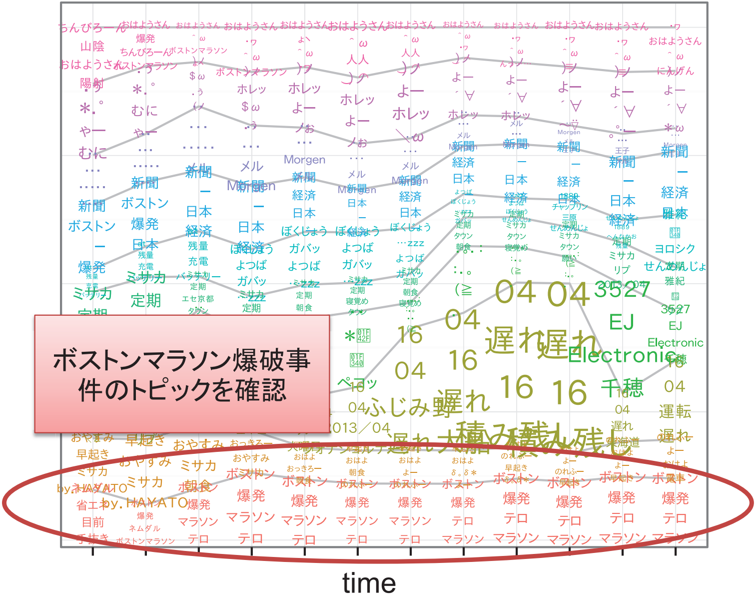
	NMF	1%	10%	100%
Batch		9.01E+09	6.32E+06	4.51E+04
Online	[Cao+ 07]	1.06E+04	3.25E+04	1.83E+04
Dynamic	[Saha+ 12]	6.53E+04	N/A	N/A
Streaming	[proposed]	5.65E+07	3.25E+04	8.71E+03
w/ Filter	[proposed]	5.25E+09	2.40E+04	7.90E+03

✔ Streaming NMF: best at 100% data

✔ Topic hijacking filter improves perplexity

[1] `http://news.yahoo.co.jp/list`

Obtained Topics



ボストンマラソン爆破事件のトピックを確認

Detected Hijacking Phrases

広告系

- シネマトゥデイ 妖精 トム 映画 ハリウッド 戦闘 クルーズ 主演
- 発売 2013 情報 開催 参加 商品 入荷
- 限定 情報 開催 イベント 好評 商品 @******
- リプライ 富士山 樋口 フェスティバル 興味 早稲田 開催 現在 不問 募集 スタッフ 大

拡散系

- リフォロー アカウント 19836 フォロワー Only 希望 支援 696382 交流
- 無料 日本 拡散 希望 2013/04 リプ

ボット系

- 特徴 燃料 格闘 操縦 射撃 装甲 整備 評価 機動
- 所有 City ポイント 前回 Tweet アクセス 獲得 Intel
- 自動 だれ 宣伝 AutoTweet オートツイート Twitter 定期 設定 サイト

顔文字系

- とノ ω· ┏┓ ┃ ┏┓ ┃ ┗┛ ┗┓ ┃ ┗ ┏┛ ┛ ┃ ＿┘
- ┗━┛ ┃ぅ━┫ ━┓ Å· ┃お ●∩ ┳⊂ ┃ょ。) ┏━ 。·
- *· ·゜* :.｡ ..｡ .: *·' * '·゜ :.｡ . .｡ .:

Summary

Proposed **the streaming algorithm for Twitter topic detection**

- Works in real time
  (would handle all jp tweets in theory)
- Automatically filters spam topics

Summary

Proposed **the streaming algorithm for Twitter topic detection**

- Works in real time
  (would handle all jp tweets in theory)
- Automatically filters spam topics

Thank you!

Integrated Twitter Topic Detection System

For $t = 1, \ldots, T$:

1. Generate $\mathbf{X}^{(t)}$ from tweets $\notin$ Blacklist $O(N_t)$
2. Update $\mathbf{U}, \mathbf{V}$ by SGD $O(N_t R^2 + R^3)$
3. With some intervals,
   Detect Topic Hijacking and update Blacklist $O(J)$