Hashtag Popularity Prediction

Avirup, Soumyadeep, Srinidhi, Kushal, Abhay

IIT Kharagpur

October 11, 2017

Introduction

Goals

Problem formulation

Model

Results

Conclusion

References

Introduction

- To improve the efficacy of Twitter, prediction of hashtag flow (e.g., early detection of trending hashtags [Mathioudakis and Koudas, 2010; Chakraborty et al., 2017]) is important.

Introduction

- To improve the efficacy of Twitter, prediction of hashtag flow (e.g., early detection of trending hashtags [Mathioudakis and Koudas, 2010; Chakraborty et al., 2017]) is important.
- Thus, a large amount of research on viral marketing and information cascades [Cheng et al., 2016; Chakraborty et al., 2016; Sikdar et al., 2016] have focused on analyzing hashtag-popularity and its role in tweet propagation.

Introduction

- To improve the efficacy of Twitter, prediction of hashtag flow (e.g., early detection of trending hashtags [Mathioudakis and Koudas, 2010; Chakraborty et al., 2017]) is important.
- Thus, a large amount of research on viral marketing and information cascades [Cheng et al., 2016; Chakraborty et al., 2016; Sikdar et al., 2016] have focused on analyzing hashtag-popularity and its role in tweet propagation.
- In this project we aim to predict the popularity of hashtags by modeling the occurrence of tweets bearing the hashtag as a temporal point process.

Prior works and their limitations

Research on predictive aspects of hashtag-popularity primarily follows two kinds of models:

- Static feature-based models ([Rosenfeld et al., 2016; Bourigault et al., 2014; Shi et al., 2016])

Prior works and their limitations

Research on predictive aspects of hashtag-popularity primarily follows two kinds of models:

- Static feature-based models ([Rosenfeld et al., 2016; Bourigault et al., 2014; Shi et al., 2016])
- Temporal models ([Zhao et al., 2015; Kobayashi and Lambiotte, 2016; Farajtabar et al., 2015; Bi and Cho, 2016; De et al., 2016a; Iwata et al., 2013; De et al., 2014; Kupavskii et al., 2012; Hua-Wei et al., 2014; Shuai and Jun, 2015; Gao et al., 2016; Ferraz Costa et al., 2015; Bao et al., 2015; Gomez-Rodriguez et al., 2011; De et al., 2016b])

Prior works and their limitations

- In the static models, the temporal properties (e.g., time of posts, no. of retweets) are embedded into feature maps, and the parameters are learned following a supervised approach.

Prior works and their limitations

- In the static models, the temporal properties (e.g., time of posts, no. of retweets) are embedded into feature maps, and the parameters are learned following a supervised approach.
- However, in practice, future temporal properties are not known in advance, which in turn constrains their forecasting prowess.

Prior works and their limitations

- The temporal models aim to address this shortcoming by modeling the stochastic nature of hashtag-dynamics, using point-process, survival theory, etc.

Prior works and their limitations

- The temporal models aim to address this shortcoming by modeling the stochastic nature of hashtag-dynamics, using point-process, survival theory, etc.
- However, most existing temporal models focus on tweet-propagation rather than hashtags, thereby skirting several realistic aspects of hashtag-flow, and resulting in modest prediction performance.

Prior works and their limitations

- The temporal models aim to address this shortcoming by modeling the stochastic nature of hashtag-dynamics, using point-process, survival theory, etc.
- However, most existing temporal models focus on tweet-propagation rather than hashtags, thereby skirting several realistic aspects of hashtag-flow, and resulting in modest prediction performance.
- More importantly, they are largely unable to reproduce any microscopic feature in hashtag dynamics (e.g., relative popularity variation, sudden trend change etc.).

Goals

- To model hashtag popularity as a temporal point process in terms of popularity of tweet-chains.

Goals

- To model hashtag popularity as a temporal point process in terms of popularity of tweet-chains.
- To build a neural network based framework to predict hashtag popularity based on the above model.

Goals

- To model hashtag popularity as a temporal point process in terms of popularity of tweet-chains.
- To build a neural network based framework to predict hashtag popularity based on the above model.
- To test the model on real-world datasets.

Problem formulation

- We model the conditional intensity, $\lambda(t)$ of the temporal point process in terms of the tweet-popularity index k and the event history of the point process.

Problem formulation

- We model the conditional intensity, $\lambda(t)$ of the temporal point process in terms of the tweet-popularity index k and the event history of the point process.
- We use this conditional intensity to predict the expected number of tweets bearing the same hashtag posted in a subsequent time interval.

Problem formulation

- We model the conditional intensity, $\lambda(t)$ of the temporal point process in terms of the tweet-popularity index k and the event history of the point process.
- We use this conditional intensity to predict the expected number of tweets bearing the same hashtag posted in a subsequent time interval.
- We evaluate the accuracy of the prediction by comparing our predicted value with the ground truth.

Definitions

- Formally, a tweet-chain C_i, corresponding to tweet i, can be written as $C_i = \{t_j | \text{tweet } i \text{ is (re)tweeted at time } t_j\}$.

Definitions

- Formally, a tweet-chain C_i, corresponding to tweet i, can be written as $C_i = \{t_j | \text{tweet } i \text{ is (re)tweeted at time } t_j\}$.
- Similarly, a hashtag $\boldsymbol{H}$ can be expressed in terms of the tweet-chains, $\boldsymbol{H} = \{C_i | C_i \text{ is a tweet-chain bearing } \boldsymbol{H}\}$.

Definitions

- Formally, a tweet-chain C_i, corresponding to tweet i, can be written as $C_i = \{t_j | \text{tweet } i \text{ is (re)tweeted at time } t_j\}$.
- Similarly, a hashtag $\boldsymbol{H}$ can be expressed in terms of the tweet-chains, $\boldsymbol{H} = \{C_i | C_i \text{ is a tweet-chain bearing } \boldsymbol{H}\}$.
- Finally, the *history* of the hashtag $\boldsymbol{H}$ until and excluding time t, $\mathcal{H}_{\boldsymbol{H}}(t)$ can be represented as the union of the posting times of the corresponding (re)tweets posted before t.

$$\mathcal{H}_{\boldsymbol{H}}(t) = \cup_{C_i \in \boldsymbol{H}} \{t_j | t_j \in C_i \text{ and } t_j < t\}$$

Definitions

- Given a hashtag $\boldsymbol{H}$, we define the counting variable as $N_{\boldsymbol{H}}(t)$, where $N_{\boldsymbol{H}}(t) \in \{0\} \cup \mathbb{Z}^{+}$ counts the number of (re)tweets posted until and excluding time t.

Definitions

- Given a hashtag $\boldsymbol{H}$, we define the counting variable as $N_{\boldsymbol{H}}(t)$, where $N_{\boldsymbol{H}}(t) \in \{0\} \cup \mathbb{Z}^{+}$ counts the number of (re)tweets posted until and excluding time t.
- The conditional probability of observing an event in infinitesimal time interval $[t, t + dt)]$ is characterized as

$$\mathbb{P}(\text{An event triggers in } [t, t + dt)|\mathcal{H}_{\boldsymbol{H}}(t)) = \lambda_{\boldsymbol{H}}(t)\ dt \quad (1)$$

$$\text{i.e., } \mathbb{E}_{dN_{\boldsymbol{H}}(t)\sim\{0,1\}}[dN_{\boldsymbol{H}}(t)|\mathcal{H}_{\boldsymbol{H}}(t)] = \lambda_{\boldsymbol{H}}(t)dt \quad (2)$$

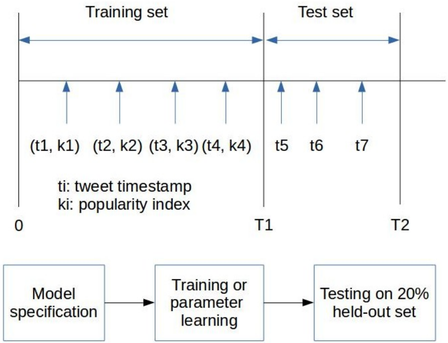
Definitions

- Here $dN_{\boldsymbol{H}}(t)$ indicates the number of (re)tweets in the infinitesimal time-window $[t, t + dt)$ and $\lambda_{\boldsymbol{H}}(t)$ stands for the associated hashtag intensity, which further depends on the history $\mathcal{H}_{\boldsymbol{H}}(t)$.

Definitions

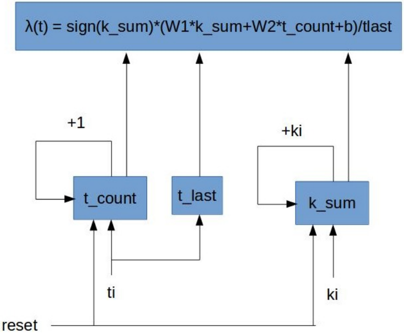
- Here $dN_{\boldsymbol{H}}(t)$ indicates the number of (re)tweets in the infinitesimal time-window $[t, t + dt)$ and $\lambda_{\boldsymbol{H}}(t)$ stands for the associated hashtag intensity, which further depends on the history $\mathcal{H}_{\boldsymbol{H}}(t)$.
- We estimate $\lambda_{\boldsymbol{H}}(t)$ using parameters learned from a given dataset using a neural network model.

Overview



We divide both training and test sets into several time intervals of equal width. Jump to conclusion

Training model



W1, W2 and b are learned by minimizing the SSE in training intervals.

Training model

We use the following features:

- **t_count:** Count of the tweets observed so far bearing a given hashtag. This value is reset to zero at the beginning of every interval.

Training model

We use the following features:

- **t_count:** Count of the tweets observed so far bearing a given hashtag. This value is reset to zero at the beginning of every interval.
- **k_sum:** Sum of the popularity indices of all tweets observed so far in the current interval bearing a given hashtag.

Training model

We use the following features:

- **t_count:** Count of the tweets observed so far bearing a given hashtag. This value is reset to zero at the beginning of every interval.
- **k_sum:** Sum of the popularity indices of all tweets observed so far in the current interval bearing a given hashtag.
- We use a modified linear combination of these two features with a bias.
- sign(k_sum) = 0 if k_sum = 0 and 1 if k_sum $>$ 0. This avoids false predictions in empty intervals.

Intuition behind Test model

- We don't know the values of **k_sum** for the test intervals.

Intuition behind Test model

- We don't know the values of **k_sum** for the test intervals.
- If the intervals are of sufficiently small width, the value of **k_sum** for the current interval will be roughly the same as the value for the last interval.

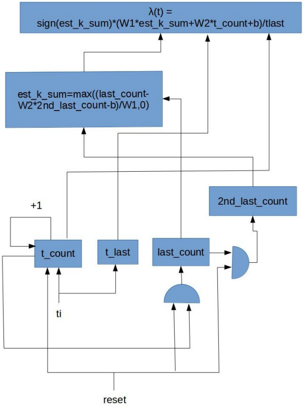
Intuition behind Test model

- We don't know the values of **k_sum** for the test intervals.
- If the intervals are of sufficiently small width, the value of **k_sum** for the current interval will be roughly the same as the value for the last interval.
- We have to estimate the value of **k_sum** for the last interval from the given data.

Intuition behind Test model

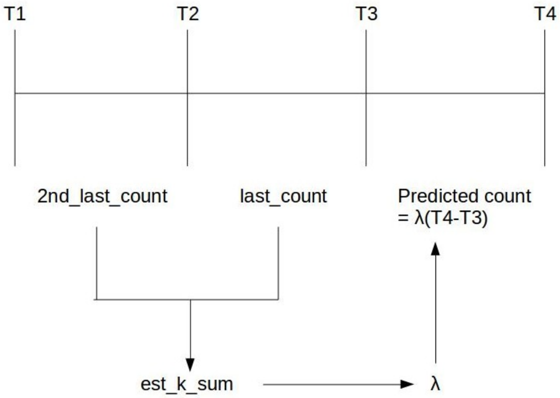
- We don't know the values of **k_sum** for the test intervals.
- If the intervals are of sufficiently small width, the value of **k_sum** for the current interval will be roughly the same as the value for the last interval.
- We have to estimate the value of **k_sum** for the last interval from the given data.
- We perform this estimation by using the learned values of the parameters together with the values of **t_count** observed for the last and 2nd last intervals.

Test model



Here W1, W2 and b are fixed.

Prediction in the test set



Evaluation Metrics

- **Mean Absolute Percentage Error (MAPE):** It captures the mean deviation between the observed and the predicted popularity for a hahstag up to time t. It is defined by the formula

$$\mathrm{MAPE}(\boldsymbol{H}) = \frac{1}{M_{\boldsymbol{H}}} \sum_{i=0}^{M_{\boldsymbol{H}}-1} \left| \frac{\hat{N}_{\boldsymbol{H}}(t_i) - N_{\boldsymbol{H}}(t_i)}{N_{\boldsymbol{H}}(t_i)} \right| .$$

Evaluation Metrics

- **Mean Absolute Percentage Error (MAPE):** It captures the mean deviation between the observed and the predicted popularity for a hahstag up to time t. It is defined by the formula

$$\text{MAPE}(\boldsymbol{H}) = \frac{1}{M_{\boldsymbol{H}}} \sum_{i=0}^{M_{\boldsymbol{H}}-1} \left| \frac{\hat{N}_{\boldsymbol{H}}(t_i) - N_{\boldsymbol{H}}(t_i)}{N_{\boldsymbol{H}}(t_i)} \right| .$$

- Here $\hat{N}_{\boldsymbol{H}}(t)$ and $N_{\boldsymbol{H}}(t)$ are the estimated and actual number of retweets for a hashtag $\boldsymbol{H}$ respectively, at time t. $M_{\boldsymbol{H}}$ denotes the total number messages of hashtag $\boldsymbol{H}$ in the test-set.

Evaluation Metrics

- **Spearman's Rank Correlation Coefficient (SRCC):** The Spearman's Rank Correlation Coefficient between predicted rank-list $\hat{R}_{\mathbb{H}}$, and actual rank-list $R_{\mathbb{H}}$ of a hashtag set $\mathbb{H}$ can be defined as,

$$\rho(\hat{R}_{\mathbb{H}}, R_{\mathbb{H}}) = \frac{\textbf{Cov}(\hat{R}_{\mathbb{H}}, R_{\mathbb{H}})}{\sqrt{\textbf{Var}(\hat{R}_{\mathbb{H}})\textbf{Var}(R_{\mathbb{H}})}}.$$

Evaluation Metrics

- **Spearman's Rank Correlation Coefficient (SRCC):** The Spearman's Rank Correlation Coefficient between predicted rank-list $\hat{R}_{\mathbb{H}}$, and actual rank-list $R_{\mathbb{H}}$ of a hashtag set $\mathbb{H}$ can be defined as,

$$\rho(\hat{R}_{\mathbb{H}}, R_{\mathbb{H}}) = \frac{\textbf{Cov}(\hat{R}_{\mathbb{H}}, R_{\mathbb{H}})}{\sqrt{\textbf{Var}(\hat{R}_{\mathbb{H}})\textbf{Var}(R_{\mathbb{H}})}}.$$

- Here **Cov**(.) defines covariance of the two variables, and **Var**(.) denotes the variance.

Evaluation Metrics

- **Avg. Recall (AvRe) and Avg. Precision (AvPr) (for jump detection):** If in two consecutive time-intervals there is a sudden change in the rank of a hashtag by more than half the total no. of competing hashtags, we call it a *jump*.

Evaluation Metrics

- **Avg. Recall (AvRe) and Avg. Precision (AvPr) (for jump detection):** If in two consecutive time-intervals there is a sudden change in the rank of a hashtag by more than half the total no. of competing hashtags, we call it a *jump*.
- Let a hashtag $\boldsymbol{H} \in \mathbb{H}$ have a rank $\text{rank}_{\boldsymbol{H},[t_i,t_{i+1})}$ in the time-interval $[t_i, t_{i+1})$. Then, if $|\text{rank}_{\boldsymbol{H},[t_i,t_{i+1})} - \text{rank}_{\boldsymbol{H},[t_{i-1},t_i)}| \geq |\mathbb{H}|/2$, it is considered a jump.

Evaluation Metrics

- **Avg. Recall (AvRe) and Avg. Precision (AvPr) (for jump detection):** If in two consecutive time-intervals there is a sudden change in the rank of a hashtag by more than half the total no. of competing hashtags, we call it a *jump*.
- Let a hashtag $\boldsymbol{H} \in \mathbb{H}$ have a rank $\text{rank}_{\boldsymbol{H},[t_i,t_{i+1})}$ in the time-interval $[t_i, t_{i+1})$. Then, if $|\text{rank}_{\boldsymbol{H},[t_i,t_{i+1})} - \text{rank}_{\boldsymbol{H},[t_{i-1},t_i)}| \geq |\mathbb{H}|/2$, it is considered a jump.
- Recall measures the fraction of cases in which an algorithm rightly identifies a jump, while precision measures the fraction of cases where a real jump has occurred when the algorithm predicts a jump.

Datasets

- Seven datasets were created by carefully selecting a few hashtags from raw data obtained using Twitter search API to collect all the tweets related to the following events (corresponding to a 2-3 weeks period around the event date):

Datasets

- Seven datasets were created by carefully selecting a few hashtags from raw data obtained using Twitter search API to collect all the tweets related to the following events (corresponding to a 2-3 weeks period around the event date):
- Events: (i) *The Academy Awards 2016* (Oscars), (ii) *MTV Awards 2016* (MTV), (iii) *Earthquake in Nepal 2015* (Nepal-Earthquake), (iv) *Democratic Primaries for US Presidential Election 2016* (Dem-Primary), (v) *Big Billion Day sale of e-commerce site Flipkart 2014* (BBD), (vi) *Copa America Football Tournament 2016* (Copa), and (vii) *T20 Cricket World Cup 2016* (T20WC).

MAPE and SRCC Data

	MAPE(%)							SRCC						
Datasets	**New**	**LMPP**	**HTR**	**RPP**	**Hawkes**	**SEISMIC**	**SpikeM**	**New**	**LMPP**	**HTR**	**RPP**	**Hawkes**	**SEISMIC**	**SpikeM**
Oscars	**15.90**	**18.90**	21.78	24.30	19.23	24.79	27.11	**0.86**	**0.85**	0.68	0.80	0.52	0.10	0.74
MTV-Awards	**4.46**	**05.14**	06.76	15.37	13.57	19.09	24.45	**0.97**	**0.87**	0.86	0.81	0.75	0.70	0.82
Nepal-Earthquake	**6.72**	**07.50**	08.54	22.28	15.42	13.73	17.95	**0.92**	**0.91**	0.87	0.63	0.32	0.63	0.75
Dem-Primary	**6.68**	**08.33**	10.50	11.33	11.62	26.09	19.12	**0.92**	**0.86**	0.68	0.80	0.51	0.10	0.73
BBD	**12.86**	**15.40**	17.94	19.09	15.94	18.03	20.89	**0.96**	**0.95**	0.79	0.90	0.91	0.43	0.79
Copa	**11.59**	**17.67**	20.07	17.75	19.18	23.32	22.44	**0.96**	**0.91**	0.42	0.75	0.88	0.42	0.64
T20WC	**9.85**	**10.25**	11.90	13.10	15.08	25.55	41.74	**0.98**	**0.87**	0.58	0.83	0.31	0.56	0.47

The cells with yellow (blue) color indicate the best (second best) predictor.

Jump Detection Statistics

	Avg. Precision							Avg. Recall						
Datasets	**New**	**LMPP**	**HTR**	**RPP**	**Hawkes**	**SEISMIC**	**SpikeM**	**New**	**LMPP**	**HTR**	**RPP**	**Hawkes**	**SEISMIC**	**SpikeM**
Oscars	**0.62**	**0.74**	0.54	0.32	0.38	0.31	0.33	**0.80**	**0.75**	0.45	0.32	0.37	0.33	0.34
MTV-Awards	**0.86**	**0.31**	0.30	0.30	0.31	0.31	0.30	**0.75**	**0.33**	0.32	0.33	0.33	0.30	0.32
Nepal-Earthquake	**0.78**	**0.61**	0.60	0.37	0.28	0.40	0.44	**0.92**	**0.70**	0.54	0.37	0.33	0.52	0.57
Dem-Primary	**0.73**	**0.69**	0.56	0.48	0.30	0.45	0.34	**1.0**	**0.72**	0.49	0.48	0.29	0.57	0.36
BBD	**0.89**	**0.66**	0.48	0.32	0.55	0.31	0.43	**1.0**	**0.67**	0.48	0.32	0.64	0.28	0.40
Copa	**1.0**	**0.72**	0.29	0.42	0.60	0.29	0.34	**0.88**	**0.59**	0.33	0.42	0.53	0.33	0.42
T20WC	**0.75**	**1.0**	0.32	0.64	0.10	0.29	0.65	**1.0**	**0.67**	0.32	0.64	0.10	0.21	0.54

In general our model has high recall but it tends to make false jump predictions.

Conclusion

Jump to overview

- We have proposed a linear neural-network based framework to model the hashtag flows as a temporal point process.

Conclusion

Jump to overview

- We have proposed a linear neural-network based framework to model the hashtag flows as a temporal point process.
- We achieve better results than all baselines in almost all datasets.

Conclusion

Jump to overview

- We have proposed a linear neural-network based framework to model the hashtag flows as a temporal point process.
- We achieve better results than all baselines in almost all datasets.
- This model can perhaps be improved further by introducing nonlinearity and explicitly modeling the interactions between hashtags.

References

- Samanta, Bidisha, Abir De, Abhijnan Chakraborty, and Niloy Ganguly. "LMPP: A Large Margin Point Process Combining Reinforcement and Competition for Modeling Hashtag Popularity."
- Du, Nan, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. "Recurrent marked temporal point processes: Embedding event history to vector." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1555-1564. ACM, 2016.