Time Series Analysis Using NOC

Noriaki Kawamae
The University of Tokyo
7 Chome-3-1 Hongo, Bunkyo, Tokyo, Japan
Japan 113-8654
kawamae@gmail.com

ABSTRACT

We present a time series analysis employing natural language processing (NLP) techniques, and show the effect of N-gram over Context (NOC), that is a one of topic models that enjoy success in NLP, in this analysis.

Keywords

Time series analysis, Nonparametric models, Topic models, Latent variable models, N-gram topic model

1. INTRODUCTION

Since many time series data sets look like to text data sets in having some simple patterns and the importance of their order, our analysis is based on the motivation that natural language processing (NLP) techniques could be applied to time series analysis via symbolic representation, and allow taking advantages of these techniques. While this concept seems straightforward, it requires subtleties. This paper shows how to describe the generative process of time series by using topic models that enjoy success in NLP [5].

2. OUR APPROACH

Our approach segments each time series into patterns, captures a group of similar patterns, and represents the generative process of each time series using these groups, where we mean that each pattern, and group corresponds to a phrase, and topic in NLP. Since time series has no boundaries, a dictionary of patterns and these patterns depend on a given data, we firstly transform time series into uniformly sized bins to construct a dictionary automatically from a given data.

After attaining the global minimum value, min, and the global maximum value, max, we define the bin width by $(max - min)/cardinality$, and then gain its probability, p_l, using the frequency of data points observed in l-th bin on a given time series. As shown in Figure 1, we derive a variable-length code by applying a Huffman coding [2] to these probabilities, and represent each time series as a sequence of these symbolic codes that correspond words in text data.

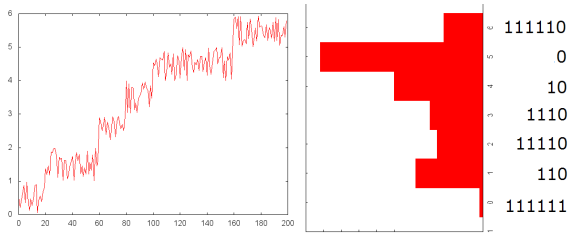

Figure 1: Time series and its bin representation with the probability for Huffman encoding

Our challenge is to detect patterns and group of similar patterns from these transformed time series data. Because the sum of the individual codes and its meaning can depend on context, and the order of codes exhibits the meaning of each time series, our proposal analysis detects phrases as patterns, and reveals topics as groups by applying a N-gram topic model to these discretized time series data. This approach represents each time series as a mixture of a low-dimensional set of group, where each group has a multinomial distribution over symbolic codes. Like other topic models, this symbolic presentation allows us to capture what time series is generally about beyond the specific codes it contains and strong to code-choice noise and match time series via groups instead of codes. Since both time series and languages are intrinsically cohesive and coherent, modeling group specific patterns would be also beneficial in time series analysis.

N-gram over Context (NOC) [4] takes a Bayesian nonparametric approach to automatically estimate the appropriate number of topics, which impacts the quality of topic specific words, N-grams, and observed value distribution. Since NOC reveals a tree of topics that captures the semantic relationship between topics from a given corpus, and forms N-gram by offering power-law distributions for word frequencies on this topic tree to improve the quality of topic specific N-grams, our proposal analysis employs NOC as N-gram topic model. Figure 2 shows the graphical model of NOC, and Table 1 shows its notations.

3. EXPERIMENTS

To help our interpretation, we evaluate this approach us-

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
Include the http://dx.doi.org/10.1145/2872518.2889396.

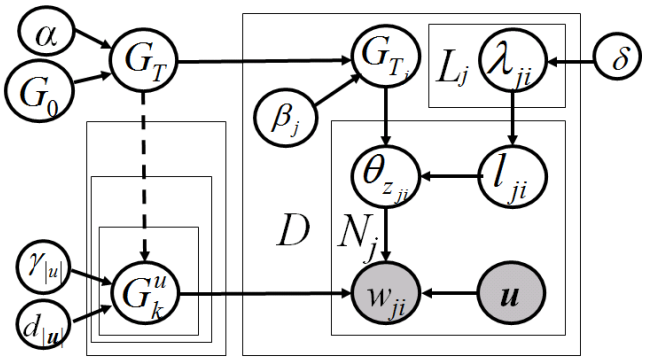

Figure 2: Graphical Model of NOC: In this figure, shaded and unshaded variables indicate observed and latent variables, respectively. An arrow indicates a conditional dependency between variables and stacked panes indicate a repeated sampling with the iteration number shown. $\theta_{z_{ji}}$ is associated with one ϕ_k via the topic indicator, k.

ing Twitter data[1], where we selected tweets from 30/08/2012 to 29/01/2013, and gained a corpus consisting of 65,456,988 users and 321,513,597 tweets. From this data set, we collected tweets including top 1000 most frequently mentioned hashtags and made latest 2-month time series for each hash tag, j, by counting the number of tweets on each day, w_{ji}, and aligning them on each hash tag. We set the cardinality of this data to 1000, and ran the experiments on 40 PCs with Hadoop[2] and Dual Core 2.66 GHz Xeon processors and the number of Gibbs sampling iterations was set to 1000.

Firstly, we used K-means clustering to evaluate the effect of coded representation, and phrases in the classification task, and measured the average cluster quality of them under K=20. Table 2 shows that phrases representation could reduce sensitivity to noise and fits for time series analysis, since clusters using phrases attain similarity values 1.20 times higher than clusters gained from raw time series.

Because value prediction is also one of most interesting tasks in time series data and this data is converted into codes with their probability, this task can be evaluated using the test-set perplexity (PPX), which is a standard measure in NLP to assess the predictive power of a model. To compare the generative ability of time series with HDP [5]-LDA [1], and N-gram topic models such as NTSeg [3], we computed PPX following the setting [4] using the averages of ten-fold cross validation, and show the result in Table 3. This table confirms our assumption that NOC could be applied to time series analysis through the appropriate coding approach, and could fit for modeling and explaining the generative process of time series, since it shows the lowest PPX.

4. CONCLUSION

Our contribution lies in evaluating and showing the effect of NLP techniques in the time series analysis. Future work is to extend this approach to reveal significant correlations between different time series.

[1]Twitter: http://twitter.com
[2]Apache™Hadoop®: http://hadoop.apache.org/

Table 1: Notation used in this paper

SYMBOL	DESCRIPTION
$D(W)$	♯documents (vocabulary size)
$N_j(L_j)$	♯words (topic level)in j-th document
$z_{ji}(l_{ji})$	the i-th topic (level) variable in j-th document
w_{ji}	the i-th word in j-th document
h_{ji}	the previous word sequence sharing with the same topic before w_{ji}
G_0	a $\|W\|$-dimensional uniform word distribution
$G_T(G_{T_j})$	the topic distribution of T (T_j)
$G_{T^l}(G_{T_j^l})$	the l-th level topic distribution of T (T_j)
$\phi_{l,p,c}(\phi_k)$	topic: the c-th child in l-th level of p-th parent topic in l-1 level (shorthand of $\phi_{l,p,c}$)
$\mathbf{u}$	a n-1 words sequence sharing the same topic
$G_k^{\mathbf{u}}$	the topic k specific word distribution following $\mathbf{u}$
$d_{\|\mathbf{u}\|}(\gamma_{\|\mathbf{u}\|})$	the $\|\mathbf{u}\|$ specific discount (concentration): parameter $d_{\|\mathbf{u}\|} \sim Beta(e_{\|\mathbf{u}\|}, f_{\|\mathbf{u}\|})$ $(\gamma_{\|\mathbf{u}\|} \sim Gamma(g_{\|\mathbf{u}\|}, h_{\|\mathbf{u}\|}))$
λ_{jl}	the l-th topic level of T_j specific beta random: variable $\lambda_{jl} \sim Beta(\delta_1, \delta_2)$
β_j	the j specific parameters: $\beta_{jl} \sim Gamma(a_1, a_2)$

Table 2: Performance quality comparison over time series with various representation: #topics is automatically defined as 57 from a given data. Code1, code2, and code3 means the code representation using NOC with only unigram, unigram+2-gram (N<3), and unigram+2,3-gram (N<4). Results that differ significantly, t-test $p < 0.01$, from the "raw" are marked with '*'.

data representation	raw	code1	code2	code3
average accuracy	0.483	0.612*	0.633*	0.652*

Table 3: PPX comparison over symbolic presentation time series: #topics of NTSeg is in tune with that of HDP-LDA. Results that differ significantly, t-test $p < 0.01$, from the "NTSeg".

topic model	HDP-LDA	NTSeg	NOC
PPX	1324	1311	1123

5. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[2] D. A. Huffman. A method for the construction of minimum-redundancy codes. In *Proceedings of the Institution of Radio Engineers*, pages 1098–1101, 1952.

[3] S. Jameel and W. Lam. An unsupervised topic segmentation model incorporating word order. In *SIGIR*, pages 203–312, 2013.

[4] N. Kawamae. N-gram over context. In *WWW, to appear*, 2016.

[5] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *JASA*, 101(476):1566–1581, 2006.