Accelerating Topic Detection on Web for a Large-Scale Data Set via Stochastic Poisson Deconvolution

Jinzhong Lin[1], Junbiao Pang[2], Li Su[1], Yugui Liu[1], and Qingming Huang[1,3](✉)

[1] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
lin_jin_zhong@outlook.com, {suli,liuyg,qmhuang}@ucas.ac.cn
[2] Faculty of Information Technology, Beijing University of Technology, Beijing, China
junbiao_pang@bjut.edu.cn
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Abstract. Organizing webpages into hot topics is one of the key steps to understand the trends from multi-modal web data. To handle this pressing problem, Poisson Deconvolution (PD), a state-of-the-art method, recently is proposed to rank the interestingness of web topics on a similarity graph. Nevertheless, in terms of scalability, PD optimized by expectation-maximization is not sufficiently efficient for a large-scale data set. In this paper, we develop a Stochastic Poisson Deconvolution (SPD) to deal with the large-scale web data sets. Experiments demonstrate the efficacy of the proposed approach in comparison with the state-of-the-art methods on two public data sets and one large-scale synthetic data set.

Keywords: Large-scale · Poisson Deconvolution · Unsupervised Ranking · Web Topic Detection · Surrogate Function.

1 Introduction

With the rapid development of information technology and mobile internet, social media websites greatly facilitate both the generation and the propagation of User-Generated Content (UGC). Consequently, the unprecedented explosion in the volume of UGC [10] data makes people difficult to quickly grasp "hot" contents. Driven by this practical requirement, topic detection from web [10, 26, 11] is such an effort to organize webpages into meaningful topics automatically. Different from the traditional Topic Detection and Tracking (TDT) [5] that aims at discovering topics from professionally edited news articles, web topic detection faces a large-scale UGC data which never evolve into any hot topics. In this paper, web topic detection is formally defined as the task of discovering of a tiny fraction of webpages strongly connected by a seminal hot event from a large amount of social media [10].

The state-of-the-art approach for web topic detection is to rank the interestingness of topics on a similarity graph [10, 12]. Concretely, PD allocates an weight to each topic by diffusing the similarities between webpages [10]. Although a similarity graph is not only efficiently constructed by online k-Nearest Neighborhood Graph (k-N^{2}G) [23] but also is efficiently stored as a sparse matrix, one pressing problem is that PD is not efficiently scalable for a large-scale data set. The reason is that PD has to reconstruct a $N \times N$ float matrix at each iteration where N is the number of webpages.

It is natural to ask: *can we exploit a small fraction of data at each iteration for PD?* One of the simple and yet efficient approaches is stochastic optimization [14]. There are at least two potential benefits of this approach: reducing the requirement of the physical memory, and avoiding the reconstruction of a $N \times N$ scale similarity graph. However, PD optimized by EM algorithm has to maintain a hidden variable which has the same scale of the similarity graph.

In this paper, we propose a Stochastic Poisson Deconvolution (SPD) approach to handle a large-scale data set for web topic detection. It iteratively builds a surrogate of the expected objective function when only a small fraction of data are observed at each iteration. Meanwhile, only a few small stochastically sampled data are used to update the surrogate function. Therefore, avoiding loading all data into memory, SPD significantly reduces the running time.

To the best of our knowledge, this is the first to handle the scalability of PD for web topic detection. The proposed method is conceptually simple and yet efficiently. On a large-scale data set, SPD leads to drastic training-time improvement, *e.g.*, approximate 12.6× speedup on a toy data set with about 200,000 webpages. Meanwhile, SPD can achieve the similar performances to that of PD on two public data sets.

The rest of this paper is organized as follows: Section 2 reviews the related work. We describe the details of our approach in Section 3. Experimental results are presented in Section 4, and the paper is concluded in Section 5.

2 Related Work

Detect Web Topic from Multi-modal Data. Recognizing that webpages are the typical heterogeneous data, many literatures consider web topic detection as the clustering task from the multi-modal data. There are two important research threads. One is the multi-modal-based method [2, 4], and the other is the similarity graph-based method [22].

In the former thread, topic detection extends the single-modality based approaches into multi-modal data. For instance, multi-modal LDA [4], a variation of LDA [3], is proposed to detect topics from both the images and their tags. In the similarity graph-based method, multi-modal information is fused into the edges of a graph, where the vertexes are clustered into different topics. For instance, Wu *et al.* [24] detect topics of news videos by fusing the similarities from Nearly-Duplicated Keyframes (NDKs) and the speech transcripts.

Compared with the multi-modal-based topic modelings [4], similarity graph-based approach is easily extendable for the other algorithms [16, 22, 6]. SPD belongs to the similarity graph-based approach. However, our method does not aim at improving accuracy of the detection system, but rather making PD scalable for a large scale data set.

Despite many approaches propose to detect topics in social media, to the best of our knowledge, only a few solutions try to parallelize LDA, *e.g.*, [25, 7, 27]. As discussed in [10, 11], LDA assumes that each webpage should belong to a topic. In fact, in terms of web topic detection, almost 90% webpages would not evolve into any hot topics. Therefore, the paralleled LDAs are incapable to remove the low-valued webpages which do not develop into hot topics.

Stochastic Optimization. Stochastic optimization refers to the minimization (or maximization) of a function in the presence of randomness. The randomness may be presented as noises in measurements, Monte Carlo randomness during the search, or both [14]. For instance, Stochastic Gradient Descent (SGD) and its variants [8, 20, 18] has been popular in machine learning due to their efficiency and effectiveness. However, the objective function of PD is iteratively changed at each expectation step. This makes SGD unusable for the EM-based PD.

Majorization-Minimization (MM) [13], a generalization of EM [19, 21], iteratively minimizes a surrogate function that is the upper bound of the objective function. Many approaches can be interpreted as MM, such as variational Bayes [17] and proximal algorithms [1]. Recently, Stochastic MM (SMM) [9] is proposed to make MM scalable.

Inspired by the success of warm restart and SMM [9], our proposed method additively updates a surrogate function. The resulting SPD not only stores a few random sampled edges, but also significantly speeds up the convergence speed in practice. To the best of our knowledge, this paper is first to apply the surrogated-based method to accelerate PD for web topic detection.

3 Stochastic Poisson Deconvolution

3.1 Revisit Poisson Deconvolution

Given a set of webpages $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we convert these webpages into a similarity graph $G = (V, E, A)$, where the vertex set V corresponds to the webpages $\mathcal{X}$, the elements of affinity matrix A ($a_{ij} \in A$) corresponds to the scaled and truncated similarities between webpages $\mathcal{X}$, and the edge set E ($e_{ij} \in E$) corresponds to the similarities between webpages $\mathcal{X}$. The details about how to build a similarity graph can be founded in [11].

A set of multi-granularity topics C_k ($k = 1, \ldots, K$) are generated from a similarity graph G, where a topic C_k is represented as:

$$C_k = c_k^\top \circ c_k, \tag{1}$$

in which the indicator vector $c_k \in \{0, 1\}^{1 \times N}$, where 1 or 0 means that whether the topic C_k contains the webpage $\mathbf{x}_i$ or not. The operation $\circ$ means that the diagonal of the matrix $c_k^\top c_k$ is set to be zero.

Given a set of topics $\{C_1, \ldots, C_K\}$ and a similarity Graph $G = (V, E, A)$, the topic-wise weight μ_k of a topic C_k is learned under Poisson noise as follows:

$$\begin{aligned} w_{ij} &\sim \text{Poisson}(a_{ij}) \\ \textit{where} \;\; w_{ij} &= \sum_{k=1}^{K} \mu_k C_{k_{ij}}. \end{aligned} \tag{2}$$

The interestingness of a topic is estimated as $i_k = \mu_k \cdot |C_k|$, where $|C_k|$ is the number of webpages in the topic C_k.

By applying EM algorithm, PD (2) is iteratively solved as follows:

$$\mu_k = \frac{\sum_{a_{ij} \in C_k} a_{ij} P_{k_{ij}}}{\sum_{a_{ij} \in C_k} C_{k_{ij}}}, \tag{3}$$

where $P_{k_{ij}}$ ($\sum_{k=1}^{K} P_{k_{ij}} = 1$) are the hidden variables, i.e., $P_{k_{ij}} = \frac{\mu_k C_{k_{ij}}}{\sum_{m=1}^{K} \mu_m C_{m_{ij}}}$ [10].

The Drawback of Poisson Deconvolution: Eqt. (3) needs to reconstruct all edges in a similarity graph G at each iteration. In practice, a float $N \times N$ matrix has to be allocated in memory; besides, at each iteration, each element of the $N \times N$ matrix has to be updated. Therefore, the time complexity of PD is $O(TN^2)$ where T is the number of iterations; meanwhile, the space complexity of PD is $O(N^2)$. The polynomial complexity handicaps the scalability of PD for a large-scale data set. This problem looms as long as the computation-intensive $N \times N$ matrix is required to be allocated and reconstructed.

3.2 Stochastic Poisson Deconvolution

We present SPD for a large-scale data set in Algorithm 1. At each iteration, by assuming that the edges in a similarity graph are *i.i.d.* from an unknown distribution, we draw a mini-batch edges $\bar{A}^t$ from a similarity graph at the t-th iteration. However, in practice, the mini-batch edges are computed by cycling on a randomly permuted training set [15], since it is often difficult to obtain true *i.i.d.* samples.

Concretely, the objective function of PD (2) is equal to the following problem:

$$\begin{aligned} &\max \ln \prod_{a_{ij} \in \bar{A}^t} \text{Poisson}(a_{ij}) \\ \Leftrightarrow &\min \underbrace{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} \Big(\sum_{k=1}^{K} \mu_k C_{k_{ij}} - a_{ij} \ln \sum_{k=1}^{K} \mu_k C_{k_{ij}} \Big)}_{f^t(\bar{A}^t, \boldsymbol{\mu})}, \end{aligned} \tag{4}$$

where b is the number of rows in a mini-batch $\bar{A}^t (\bar{A}^t \in \mathbb{R}^{b\text{x}\mathrm{N}})$.

```
Algorithm 1: Stochastic Poisson Deconvolution
  Input: $G$ (Similarity Graph), $C_k$ $k = 1, \ldots, K$ (Topics), $b$ (Batch size), $T$
         (Number of iteration);
  Initialization: cumulative intermediates:$\bar{B}^0 = \bar{D}^0 = 0^{K\text{x}1}$; $W = 0^{K\text{x}1}$; $\beta$, $\alpha$;
   $\boldsymbol{\mu}^0$;
  for $t = 1, \ldots, T$ do
      Randomly draw a few rows of G: $\bar{A}^t$;
      compute the weight by (9);
      compute the temporary variable $B_k^t$, $D_k^t$ by (6);
      update the cumulative intermediates:
          $\bar{B}_k^t = (1 - w_k^t)\bar{B}_k^{t-1} + w_k^t B_k^t$;
          $\bar{D}_k^t = (1 - w_k^t)\bar{D}_k^{t-1} + w_k^t D_k^t$;
      update the current estimate: $\mu_k^t = \frac{\bar{B}_k^t}{\bar{D}_k^t}$;
  end
  Output: $\boldsymbol{\mu}$;
```

Using Jensen's inequality, the upper bound of likelihood function in (4) is used as the surrogate function:

$$f^t(\bar{A}^t, \boldsymbol{\mu}) \leqslant \underbrace{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} \left(\sum_{k=1}^{K} \mu_k C_{k_{ij}}^t - a_{ij} \sum_{k=1}^{K} P_{k_{ij}}^t \ln \frac{\mu_k C_{k_{ij}}^t}{P_{k_{ij}}^t} \right)}_{J^t(f^t, \boldsymbol{\mu}^{t-1})}. \tag{5}$$

Where $C_{k_{ij}}^t$ means the k-th topic contracted by sampled webpages, $P_{k_{ij}}^t$ $(\sum_{k=1}^K P_{k_{ij}}^t = 1)$ is the hidden variable for the t-th iteration, *i.e.*, $P_{k_{ij}}^t = \frac{\mu_k^{t-1} C_{k_{ij}}^t}{\sum_{k=1}^K \mu_k^{t-1} C_{k_{ij}}^t}$.

The gradient of $J^t(f^t, \mu^{t-1})$ with respect to μ_k is as follows:

$$\frac{d}{d\mu_k} J^t(f^t, \boldsymbol{\mu}^{t-1}) = \underbrace{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} C_{k_{ij}}^t}_{D_k^t} - \underbrace{\frac{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} a_{ij} P_{k_{ij}}^t}{\mu_k}}_{B_k^t}, \tag{6}$$

where $D_k^t \in \mathbb{R}^{1\text{x}1}$ and $B_k^t \in \mathbb{R}^{1\text{x}1}$ are the temporal variables.

Proposition 1 *(Iterative Update Process) Given the temporal variables B_k^t and D_k^t, μ_k can be iteratively updated as follows:*

$$\mu_k^t = \frac{\bar{B}_k^t}{\bar{D}_k^t}, \quad s.t.: \quad k \in \{k | \exists C_{k_{ij}}^t \neq 0\} \tag{7}$$

where

$$\begin{aligned} \bar{B}_k^t &\leftarrow (1 - w_k^t)\bar{B}_k^{t-1} + w_k^t B_k^t, \\ \bar{D}_k^t &\leftarrow (1 - w_k^t)\bar{D}_k^{t-1} + w_k^t D_k^t, \end{aligned} \tag{8}$$

where the weight w_k^t for the t-th iteration is as follows:

$$w_k^t = \beta \sqrt{\frac{1+\alpha}{W_k+\alpha}}, \tag{9}$$

in which $W_k = W_k + 1, \quad \beta \in (0,1], \qquad \alpha \geq 0.$

Proof. Following the suggestions in SMM [9], the combination of the approximate surrogate and the current estimation is as follows :

$$\bar{J}_k^t \leftarrow (1 - w_k^t)\bar{J}_k^{t-1} + w_k^t J_k^t, \tag{10}$$

μ_k is optimized by minimizing (10), *i.e.*, $\mu_k^t = \arg\min \bar{J}_k^t(\mu)$. This process can be derived like following:

When $t = m$, solving for μ_k^m in (10) by (6) (7) (8) can get

$$\mu_k^m = \frac{(1-w_k^m)\bar{B}_k^{m-1} + w_k^m B_k^m}{(1-w_k^m)\bar{D}_k^{m-1} + w_k^m D_k^m} = \frac{\bar{B}_k^m}{\bar{D}_k^m}. \tag{11}$$

Finally, without loss of generality, we can get μ_k^t in the iterative update process (7) (8).

The weight w plays a role of Exponentially Weighted Moving Average (EWMA) in approximate surrogate (10). The EWMA is a type of infinite impulse response filter that represents the exponentially decreased weighting factor. Note that the weight of each older surrogate decreases exponentially but never reaches to zero. The w reflects the importance of the current surrogate; meanwhile $1-w$ reflects the importance of the previous surrogates. Consequently, the older the surrogate is, the smaller the weight should be.

4 Experiments and Discusses

4.1 Data Sets, Features and Evaluation Criteria

Datasets: In the experiments, we evaluate our method on two public data sets, *i.e.*, *MCG-WEBV* [6] and *YKS* [26]. MCG-WEBV is first proposed to detect web video topics from the video sharing websites, being downloaded from the "Most viewed" videos of "This month" on *YouTube*. This data set contains video clips and their titles, tags and descriptions on *Youtube* from Dec. 2008 to Feb. 2009. YKS is a cross-media and cross-platform data set crawled from *YouKu* and *Sina*, respectively. The meta data of YKS contains news articles on *Sina* and titles, tags and descriptions web videos on *YouKu* from May 2012 to June 2012. The statistics of data sets are summarized in Table 1.

Features: During the pre-processing, MCG-WEBV and YKS are tokenized by *NLTK*[4] package. We simply use TF-IDF to encode the textual features. That

[4] www.nltk.org

Table 1. Summary of data sets in the experiments

Data sets	#Webpage	#Topics	k	Sparsity(%)
MCG-WEBV	3660	4240	100	2.73
YKS	7332	5252	20	0.273

is, the surrounding text of each video is considered as a set of words. The cosine distance is used to measure the similarity.

Evaluation Criteria: There are two evaluation methods, *i.e.*, top-10 F_1 versus Number of Detected Topics (NDT) [6] and Accuracy versus False Positive Per Topic (FPPT) [10]. Note that if two methods have the same top-10 F_1 or accuracy score, the one with smaller NDT or FPPT has better performance.

- **Top-10 F_1 versus NDT:** A detected topic is matched with the ground truth, and then the top 10 F_1 scores are averaged to measure the performance of a system:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{12}$$

  where $Precision$ is $\frac{|DT \cap GT|}{|DT|}$, $Recall$ is $\frac{|DT \cap GT|}{|GT|}$, in which DT is a detected topic, GT is a ground truth topic, and $|\cdot|$ denotes the number of webpages in a topic.
- **Accuracy versus FPPT:** if a topic is correctly detected, FPPT is the number of false positive topics caused by a detection system. In this paper, accuracy is defined as follows:

$$\text{Accuracy} = \frac{\#\text{Successful}}{\#\text{Groundtruth}}, \tag{13}$$

  where $Successful$ means a detected topic DT is successfully discovered, if Normalized Intersected Ratio (NIR) $\frac{|DT \cap GT|}{|DT \cup GT|}$ is larger than a threshold. Following the previous work [10], 0.5 is used as the threshold of NIR in our experiments.

For the time efficiency, the curve about the objective function versus the used time is used. More specially, the one with less time to reach convergence has better performance.

Enviroment Setting: To fairly compare our algorithm to the state-of-the-art method, we follow the same setting (*i.e.*, similarity graph and topics) to demonstrate the efficiency of our approach. All experiments are implemented in python with a 3.6 GHz processer and 32G RAM.

4.2 Analysis of Time Complexity and Space Complexity

For PD, a float $N \times N$ matrix has to be allocated in memory to store the reconstructed similarity graph. Therefore, the space complexity and time complexity

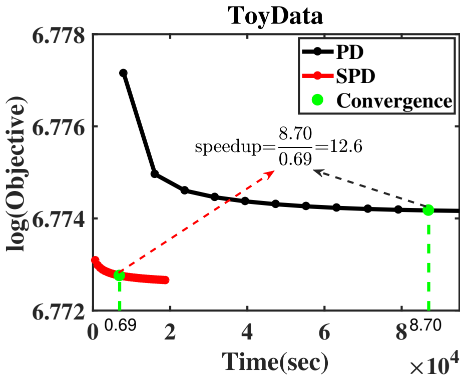
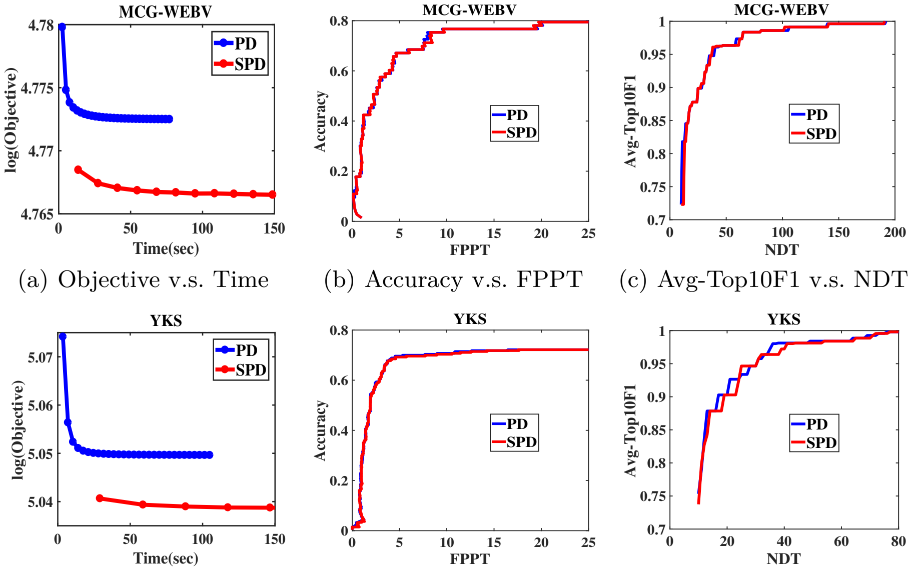


Fig. 1. Comparison between PD and SPD on ToyData (best viewed in color).



(a) Objective v.s. Time (b) Accuracy v.s. FPPT (c) Avg-Top10F1 v.s. NDT

(d) Objective v.s. Time (e) Accuracy v.s. FPPT (f) Avg-Top10F1 v.s. NDT

Fig. 2. Comparisons between PD and SPD on MCG-WEBV and YKS.

of PD is $O(N^2)$ and $O(TN^2)$, respectively. In contrast, SPD only updates a small fraction of the reconstructed similarity graph at each iteration. Therefore, SPD only maintains a float $b \times N$ matrix in memory, where $b \times N$ means the size of mini-batch edges (*i.e.*, $\bar{A}^t \in \mathbb{R}^{b\text{x}\mathrm{N}}$). The space complexity of SPD is $O(bN)$ $(b \ll N)$.

4.3 Evaluation on Toy Dataset

To evaluate the efficientness of SPD, we construct a large-scale toy data set by duplicating MCG-WEBV 50 times. The resulting toy data set has approximate $200,000$ webpages. In this paper, an algorithm converges to a local minimum, if the change of the log likelihood between two consecutive iterations is less than 10^{-5}. The speedup is a ratio of CPU time taken to converge to a local minimum between two algorithms.

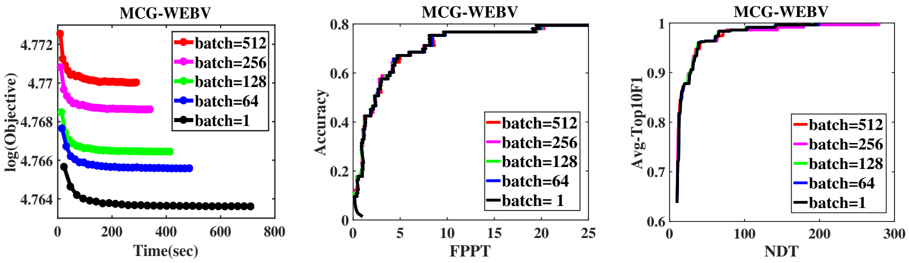
Fig. 1 shows that SPD significantly outperforms PD in terms of the convergence speed. In our evaluation, SPD achieves about 12.6× speedup. Moreover, at each iteration, although the decrease of the objective function of PD is larger than that of our method, our stochastic approach costs a fewer training time than that of the batch approach. This is because that PD uses all data to perform an accurate update while SPD only uses a small fraction of data to perform an approximate update at each iteration. When training a large-scale toy dataset, PD will cost more time than SPD with one epoch of data due to the problems of computational efficiency and memory limitations. Therefore, SPD converges much faster than PD in large-scale data sets. It validates the advantage of the loss (10): one should not spend too much effort on accurately minimizing the empirical loss. In addition, just like gradient descent, PD suffers from local minima issue while SPD can avoid it, which make SPD can converge to a smaller value than PD.

4.4 Comparisons With State-Of-The-Art Algorithms

Fig. 2 shows the comparisons between PD and SPD on both MCG-WEBV and YKS data sets. Figs. 2(b), 2(e), 2(c), and 2(f) illustrate that SPD achieves the similar performances to that of PD. Figs. 2(a) and 2(d) show that SPD can get a lower log likelihood than PD. It means that SPD can converge to a smaller value with no drop in performance. Due to these two data sets are all small, PD costs fewer time than SPD with one epoch of data. Therefore, PD converges faster than SPD in small-scale data sets.

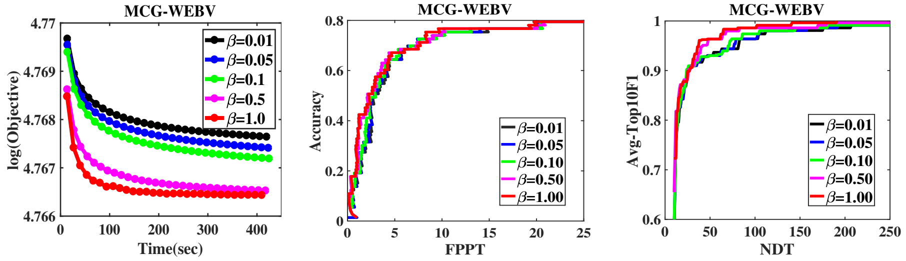
4.5 Parameter Analysis

The Size of Mini-Batch: Fig. 3 shows the effectiveness of different mini-batch size in SPD on MCG-WEBV. As shown in Fig. 3, objective values of different settings all converge to a local minimum; besides, the smaller the mini-batch is, the lower the log likelihood is. One possible reason is that a smaller mini-batch will bring a larger randomness, which may make SPD escape worse local minima and arrive at a relatively reasonable local minimum. Interestingly, the different sizes of mini-batch obtain very similar results in terms of accuracy v.s. FPPT and top-10 F_1 v.s. NDT. Therefor, the size of the mini-batch does not affect the effectiveness of SPD.



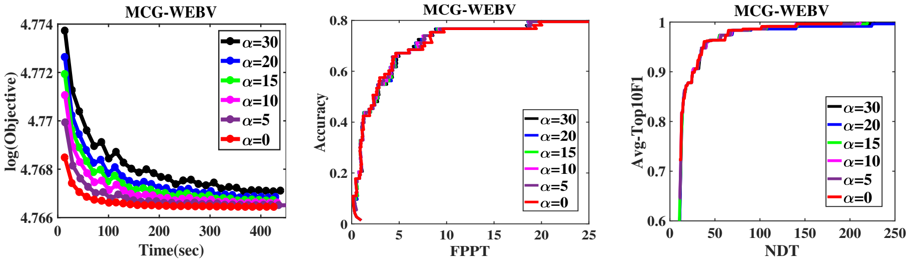
(a) Objective v.s. Time (b) Accuracy v.s. FPPT (c) Avg-Top10F1 v.s. NDT

Fig. 3. Comparisions between different batch size on MCG-WEBV.



(a) Objective v.s. Time (b) Accuracy v.s. FPPT (c) Avg-Top10F1 v.s. NDT

Fig. 4. Comparisons among different β in SPD on MCG-WEBV.



(a) Objective v.s. Time (b) Accuracy v.s. FPPT (c) Avg-Top10F1 v.s. NDT

Fig. 5. Comparison among different α in SPD on MCG-WEBV.

Weight Parameters α and β: In our implementation, we use a decreasing weight in (9), where β is a initial weight, and α is a decay factor. Fig. 4 shows comparisons among various β when other parameters are fixed, *i.e.*, α=0, mini-batch=128, and $epoch$ = 30. Fig. 4(a) illustrates the effectiveness of β on the convergence rate. As expected, the larger β is, the faster the convergence speed of SPD is. Because a larger β will make the surrogate function quickly adapt to the latest surrogate one. From Fig. 4(b) and Fig. 4(c), we find that a larger β not only results in a faster convergence speed, but also obtains a better performance.

Fig. 5 shows comparisons among various α when the other parameters are fixed, *i.e.*, β=1, mini-batch=128, $epoch$ = 30. Fig. 5(a) shows that a smaller α leads to a smoother objective function curves. Because a smaller α not only makes β decay faster, but also makes SPD stable to the latest surrogates. These different settings of α converges to a local minimum.

In summary, although different settings of α and β influence the convergence speed of SPD, both accuracy v.s. FPPT and top-10 F_1 v.s. NDT are robust to these parameters.

5 Conclusion

In this paper, we have introduced a SPD approach Sthat gracefully scales to large-scale data set for web topic detection. We have shown that our algorithm is comparable to the state-of-the-art algorithms in terms of accuracy for web topic detection. Moreover, a large-scale data set is also synthesized artificially to confirm the advantage of convergence speed. In the future, we would incorporate asynchronous parallel strategy into SPD due to the multicore systems; besides, incorporating online learning is another interesting direction.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China: 61332016, 61472389, 61672069, 61872333, 61650202 and U1636214, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013.

References

1. A.Beck, M.Teboulle: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. Siam Journal on Imaging Sciences **2**(1), 183–202 (2009)
2. D.Blei, J.Lafferty: A correlated topic model of science. Annals of Applied Sciences **1**, 17–35 (2007)
3. D.Blei, M.David, A.Ng, M.Jordan, J.Lafferty: Latent dirichlet allocation. Journal of machine learning research **3**, 993–1022 (2003)
4. D.Putthividhy, HT.Attias, SS.Magarajan: Topic regression multi-modal latent dirichlet allocation for image annotation. In: Computer Vision and Pattern Recognition. vol. 1, pp. 3408–3415 (2010)
5. J.Allan, J.Carbonell, G.Doddington, J.Yamron, et al.: Topic detection and tracking pilot study final report. In: In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. pp. 194–218 (1998)
6. J.Cao, C.Ngo, Y.Zhang, J.Li: Tracking web video topics: Discovery, visualization, and monitoring. IEEE Transactions on Circuits and Systems for Video Technology **21**(12), 1835–1846 (2011)
7. J.Chen, K.Li, J.Zhu, W.Chen: Warplda: a cache efficient o(1) algorithm for latent dirichlet allocation. Proceedings of the Vldb Endowment **9**(10), 744–755 (2015)
8. J.Mairal: Optimization with first-order surrogate functions. In: ICML (2013)
9. J.Mairal: Stochastic majorization-minimization algorithms for large-scale optimization. In: International Conference on Neural Information Processing Systems. vol. 2, pp. 2283–2291 (2013)

10. J.Pang, F.Jia, C.Zhang, W.Zhang, Q.Huang, B.Yin: Unsupervised web topic detection using a ranked clustering-like pattern across similarity cascades. IEEE Transactions on Multimedia **17**(6), 843–853 (2015)
11. J.Pang, F.Tao, C.Zhang, W.Zhang, Q.Huang, B.Yin: Robust latent poisson deconvolution from multiple features for web topic detection. IEEE Transactions on Multimedia **18**(12), 2482–2493 (2016)
12. J.Pang, F.Tao, L.Li, Q.Huang, B.Yin, Q.Tian: A two-step approach to describing web topics via probable keywords and prototype images from background-removed similarities. Neurocomputing **275**, 478–487 (2018)
13. K.Lange, DR.Hunter, I.Yang: Optimization transfer using surrogate objective functions. Journal of Computational and Graphical Statistics **9**(1), 1–20 (2000)
14. LA.Hannah: Stochastic optimization. International Encyclopedia of the Social and Behavioral Sciences **5**(5), 473–481 (2015)
15. L.Bottou, O.Bousquet: The tradeoffs of large scale learning. In: International Conference on Neural Information Processing Systems. pp. 161–168 (2007)
16. LM.Aiello, G.Petkos, C.Martin, D.Corney, S.Papadopoulos, R.Skraba, A.Göker, I.Kompatsiaris, A.Jaimes: Sensing trending topics in twitter. IEEE Transactions on Multimedia **15**(6), 1268–1282 (2013)
17. MJ.Wainwright, MI.Jordan.: Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning **1**(1–2), 1–305 (2008)
18. NL.Roux, M.Schmidt, F.Bach: A stochastic gradient method with an exponential convergence rate for finite training sets. In: International Conference on Neural Information Processing Systems. vol. 2, pp. 2663–2671 (2012)
19. O.Cappé, E.Moulines: On-line expectation-maximization algorithm for latent data models. Journal of the Royal Statistical Society **71**(3), 593–613 (2009)
20. R.Johnson, T.Zhang: Accelerating stochastic gradient descent using predictive variance reduction. In: International Conference on Neural Information Processing Systems. vol. 1, pp. 315–323 (2013)
21. RM.Neal, GE.Hinton: A view of the em algorithm that justifies incremental, sparse, and other variants. In: Nato Advanced Study Institute on Learning in Graphical MODELS. pp. 355–368 (1998)
22. S.Papadopoulous, C.Zigkolis, Y.Kompatsiaris, A.Vakali: Cluster-based landmark and event detection on tagged photo collections. IEEE Multimedia **18**(1), 52–63 (2011)
23. T.Debatty, P.Michiardi, W.Mees: Fast online k-nn graph building. CoRR (2016)
24. X.Wu, G.Hauptmann, C.Ngo: Novelty detection for crosslingual news story with visual duplicates and speech transcripts. In: ACM Multimedia. pp. 168–177 (2007)
25. Y.Wang, H.Bai, M.Stanton, W.Chen, E.Chang: Plda: Parallel latent dirichlet allocation for large-scale applications. In: Algorithmic Aspects in Information and Management. vol. 5564, pp. 301–314 (2009)
26. Y.Zhang, G.Li, L.Chu, S.Wang, W.Zhang, Q.Huang: Cross-media topic detection: a multi-modality fusion framework. In: 2013 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2013)
27. Z.Liu, Y.Zhang, EY.Chang, M.Sun: Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. ACM Trans. Intell. Syst. Technol. **2**(3), 26:1–26:18 (2011)