Soft Comput
DOI 10.1007/s00500-017-2746-9

METHODOLOGIES AND APPLICATION

SSAM: toward Supervised Sentiment and Aspect Modeling on different levels of labeling

Esmaeil Zahedi[1] · **Mohamad Saraee**[2]

Abstract In recent years, people want to express their opinion on every online service or product, and there are now a huge number of opinions on the social media, online stores and blogs. However, most of the opinions are presented in plain text and thus require a powerful method to analyze this volume of unlabeled reviews to obtain information about relevant details in minimum time and with a high accuracy. In this paper, we propose a supervised model to analyze large unlabeled opinion data sets. This model has two phases: preprocessing and a Supervised Sentiment and Aspect Model (SSAM) which is an extended version of Latent Dirichlet Allocation Model. In the preprocessing phase, we input thousands of unlabeled opinions and received a set of (key, value) pairs in which a key holds a word or an opinion and a value holds supervised information such as a sentiment label of this word or opinion. After that we give these pairs to the proposed SSAM algorithm, which incorporates different levels of supervised information such as (document and sentence) levels or (document and term) levels of supervised information, to extract and cluster aspects related to a sentiment label and also classify opinions based on their sentiments. We applied SSAM to reviews of electronic devices and books from Amazon. The experiments show that the aspects found by SSAM capture more important aspects that are closely coupled with a sentiment label, and also in sentiment classification SSAM outperforms other topic models and comes close to supervised methods.

Keywords Big unlabeled opinion dataset · Supervised Sentiment and Aspect Model · Supervised and unsupervised methods · Supervised information

Communicated by V. Loia.

✉ Mohamad Saraee
m.saraee@salford.ac.uk

Esmaeil Zahedi
e.zahedi@ec.iut.ac.ir

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

[2] School of Computing, Science and Engineering, University of Salford, Greater Manchester, UK

1 Introduction

Unsupervised extraction of Aspects from unlabeled documents is a common challenge. This challenge has been met by the topic modeling. Supervised methods (Liu et al. 2015; Poria et al. 2016) for aspect extraction are not applicable when dealing with unlabeled datasets, and they may fail when applying them on a new domain, for example, a model which learned on electronic product data is not applicable on sport domain data. Latent Dirichlet Allocation (LDA) (David et al. 2003) is more popular and has widespread use topic model. It is assumed that for each document an aspect is randomly chosen from a specified distribution, and then a word is randomly chosen according to a distribution specified by the chosen aspect. The document aspect and aspect word distributions that generate the document are unknown, but can be inferred using Gibbs sampling.

Extending these models to consider more assumptions about the data generating process makes these models more general and effective. Sentiment and topic modeling simultaneously (Lin et al. 2012; Jo and Oh 2011; Titov and McDonald 2008; Mei et al. 2007) is an informative task which is done in topic modeling-based sentiment analysis

Published online: 04 August 2017

methods. None of the existing topic models for sentiment analysis did not consider any supervised information such as review or term's sentiment label in their generative process. If we could constraint sentiments label of all words in a review be generated from one sentiment, based on review's sentiment label it would be very helpful to extract more coherent and specific aspects and also it is very useful to categorize every review in different sentiment classes. But here we have faced some limitations on real opinion datasets such as a huge number of unlabeled opinion data and lack of any knowledge about review's trend or review's sentiment. Many of works which have done in topic modeling-based sentiment analysis (Lin et al. (2012); Jo and Oh (2011); Titov and McDonald (2008); Mei et al. (2007); Poria et al. (2016); Rahman and Wang (2016); Lim and Buntine (2014)) used a little sentiment lexicons for giving sentiment label to those sentiment words that are appear in reviews. However, they could be to extract better aspects but have some problems such as extracting irrelevant aspects in different sentiment classes, having lower classification accuracy and are very time consuming due to requiring a lot of Gibbs sampling iteration to reaching a stable convergence and maybe unable to sampling on this volume of data.

In this paper, we proposed a supervised topic model called Supervised Sentiment and Aspect Model (SSAM), for classifying reviews in different sentiment classes by reformulating the generative process of LDA and adapt it to incorporate sentiment into our proposed model so that the resulting model represents the probability distributions over words for various pairs of sentiment and aspect. While aspects are drawn conditioned on review's sentiment label and words are drawn conditioned on the review's aspect and sentiment label, SSAM is capable to consider different types of supervision such as review's sentiment label and term's label where these all supervised information are calculated in preprocessing phase of the model. SSAM is distinguished from other related sentiment and topic models in its capability of accommodating with big unlabeled corpus of reviews by implementing SSAM on the big data Spark framework (Zaharia et al. 2010). We tested SSAM on the thousands of Amazon's Reviews in Electronics and Ebooks domains, and experiments results show that our proposed model significantly outperforms two strong supervised methods (SVM and NB) as well as two closed related sentiment and topic models (JST and ASUM) in sentiment classification accuracy. Aside from sentiment classification, SSAM has lower time complexity as compared to LDA and also SSAM can extract coherent and meaningful aspects. We summarize our contributions as follows:

- SSAM, which considers reviews sentiment label and terms sentiment label as the extension of LDA model by adding a sentiment layer.
- SSAM can be accounted as a full framework for classifying unlabeled reviews and cluster related words with a high accuracy.
- Our proposed model is capable of extracting implicit aspects, negation sentiments, intensified sentiments and can also consider sentence structure and terms order instead of bag of words.
- Implementation is on the big data Spark framework to adapt to the explosive growth of opinions on the web.
- A thorough analysis of the SSAM compared to other sentiment and topic modeling (e.g., JST and ASUM) and different supervised methods (e.g., SVM and NB) is presented.

The paper is organized as follows: Sect. 3 reviews some works on supervised topic models that are related to our proposed model, the SSAM and its inference procedure are described in Sect. 4, and Results and experiments on the Amazon reviews dataset are discussed in Sect. 6. Finally, conclusion and future works are outlined in Sect. 7.

2 Terminology

In this section, we define the terminology used in this paper.

- *Multiword aspect or sentiment*: an n-gram phrase that conveys aspect or sentiment, for example, "portable DVD player", "well designed".
- *negation sentiment*: a multiword with at least one sentiment word and one negation word such as *no, not, none, cannot and etc.* as the previous word, for example, "not bad", "not clear".
- *Intensified sentiment*: a multiword with at least one sentiment word and one intensified word such *as so, very, extremely and etc.* as the previous word, for example, "very well", "so expensive".
- *Aspect*: is a topic in topic modeling methods.
- *Explicit Aspect*: an aspect expression in a sentence that is a noun or noun phrase, for example, "camera", "battery".
- *Implicit Aspect*: an aspect expression in a sentence that is another type such as adjective or adverb, for example, "not fit", "expensive".
- *Sentiment Lexicons*: are the words with positive (+1) or negative (−1) sentiment, such as, good (+1) or bad (−1), which used in scoring levels of preprocessing phase.

3 Related works

Several modifications of LDA model to incorporate supervised information have been proposed in the literature. These models often involve incorporating some supervised information as prior knowledge to model learning and restriction in topic assignment. Two such types of topic modeling

depending on how the supervised information is incorporated exist. These two types are named downstream and upstream topic modeling. Downstream topic models incorporate metadata such as time, author, publication date, publication venue in their generative process where generate both word and metadata simultaneously conditioned on the topic assignment. Examples of such "downstream" models include the Topics over Time model (TOT) (Wang and McCallum 2006), CorrLDA model (Mimno and McCallum 2012), the supervised latent Dirichlet allocation (Blei and McAuliffe 2010) and Labeled LDA (Ramage et al. 2009).

The upstream topic models start with the supervised information and represent each topic as a mixture of distributions conditioned by the supervised information. Examples of the upstream type include Joint Sentiment-Topic (JST) model (Lin et al. 2012), Aspect Sentiment Unification Model (ASUM) (Jo and Oh 2011), DiskLDA (Lacoste-Julien et al. 2009), feaLDA (Lin et al. 2012), SenticLDA (Poria et al. 2016), HTSM (Rahman and Wang 2016) and TOTM (Lim and Buntine 2014). Closely related works to our proposed model are upstream topic models. In JST model, sentiment is integrated with an aspect in a single language model and sentiment and aspect words are discovered simultaneously to form a sentiment-bearing aspect, which can be used to capture sentiment association among words from different domains. Such sentiment-bearing aspect detected by JST has been used for sentiment classification. JST is a weakly-supervised model because it uses a small sentiment lexicon dataset as supervised information to modify the Dirichlet prior to sentiment-topic-word distribution. Aspect and Sentiment Unification Model (ASUM) is very similar to JST, as it extracts sentiment and aspects simultaneously by modeling each document with a sentiment distribution and a set of sentiment-specific aspect proportions. The main differences between ASUM, JST and SSAM are that both ASUM and JST do make use of a small seed set of sentiment words and have no mechanism to incorporate supervised information such as document or term sentiment labels in model inference, but SSAM can handle labeled and unlabeled data and classify unlabeled data based on the learned model. SSAM is a general model capable of operating on different levels of supervision information and works like as semi-supervised or supervised method.

FeaLDA is a supervised generative topic model for automatic detection of Web API documents form the pre-labeled web documents corpus. DiskLDA associates a single supervised label with each document and associates a topic mixture with each label; it applies a documents label transform matrix to modify Dirichlet prior of document-topic distribution in LDA model.

SenticLDA used a set of seed words, user feedback and semantic relationships between words into the model to extract more coherent aspects.

Different from the previous works where only document labels are incorporated as prior knowledge or a small sentiment lexicon used as supervised information into model learning, we propose a novel Supervised Sentiment and Aspect Model (SSAM) which is capable of incorporating supervised information derived from both the document and term sentiment labels calculated in the preprocessing phase into the generative process to constrain the model inference process and constrain the sentiment-document and sentiment–aspect-term distributions and this provides SSAM with a more stable statistical foundation.

4 Research methodology

4.1 Preprocessing phase

Raw text is usually not suitable for mining due to various reasons; hence, the raw text needs to be broken down into smaller elements such as sentences or words and also needs some preprocessing steps involving some transformations on the text. In this paper, we use different transformations including stop word removal, stemming, bigrams and n-grams extraction, implicit aspects detection, negation and intensified sentiments extraction, and the last transformation is the scoring on three different levels (term, sentence and document). Bigrams and n-grams extraction is based on approaches mentioned in Church and Hanks (1990) by applying these techniques we can find all useful n-grams, and these n-grams include almost all multiword aspects, negation and intensified sentiments.

Table 1 contains the examples of these extracted n-grams from Amazon Electronics dataset.

As shown in Fig. 1, the scoring step has three different levels based on how to spread the calculated scores into document, sentence and term levels. At the document level, the words in a document are generated from the same sentiments and aspects, in this level, a sentiment label vector, σ, is calculated according to Algorithm 3, and here we use two manually pre-defined threshold vectors *min* and *max* with length S (number of sentiments, set by user) for assigning values to σ vector elements.

For example, suppose we have three different sentiment labels (*negative*, *neutral* and *positive*), S = 3, the score value of document d is +1 and *min* and *max* vectors are $min = \{-10, -1, 1\}$, $max = \{0, 1, 10\}$, then σ_d would be: $\sigma_d = (0, 1, 1)$, this means document d has both sentiment labels 2 and 3. Output of this algorithm is document d and its sentiment vector σ. Scoring at the term level captures dependencies and neighborhoods between the words (e.g., words at left and right of a sentiment word) in a sentence and assumes a sentence may contain one or more aspects and one or more sentiment. The score value in this level is calculated by Algo-

Table 1 Extracted N-grams and their types

N-grams	Type
Digital camera	Explicit aspect
Very good	Intensified sentiment
Not good	Negation sentiment
High quality	Intensified sentiment
Very nice	Intensified sentiment
Battery life	Implicit aspect
Not waste money	Implicit aspect
External hard drive	Explicit aspect
Windows media player	Explicit aspect
Portable DVD player	Explicit aspect
Not very good	Negation sentiment
Work very well	Intensified sentiment
Not fit	Implicit aspect
Not clear	Negation sentiment
Not expensive	Implicit aspect

rithm 1. In this algorithm, $w-1$ refer to the neighbor word on the left and $w+1$ refer to the neighbor on the right of word w in a sentence. Sentence level of scoring assumes one sentence tends to represent one sentiment and one aspect.

Algorithm 2 shows the process of calculating score value at the sentence level. Output of both term and sentence level of scoring is a corpus of documents where every document has a set of (key, value) pairs.

Algorithm 1: Term level of scoring

```
FOR each document d in D
  FOR each sentence s in document d
    FOR each word w in s
      IF w in sentiment lexicons
        value(w-1) += score
        value(w) += score
        value(w+1) += score
      ENDIF
    ENDFOR
    FOR each word w in s
      Emit (w, value_w)
    ENDFOR
  ENDFOR
ENDFOR
```

Algorithm 2: Sentence level of scoring

```
FOR each document d in D
  FOR each sentence s in document d
    FOR each word w in s
      IF w in sentiment lexicons
        value(s) += score
      ENDIF
    ENDFOR
    FOR each word w in s
      Emit (w, value(s))
    ENDFOR
  ENDFOR
ENDFOR
```

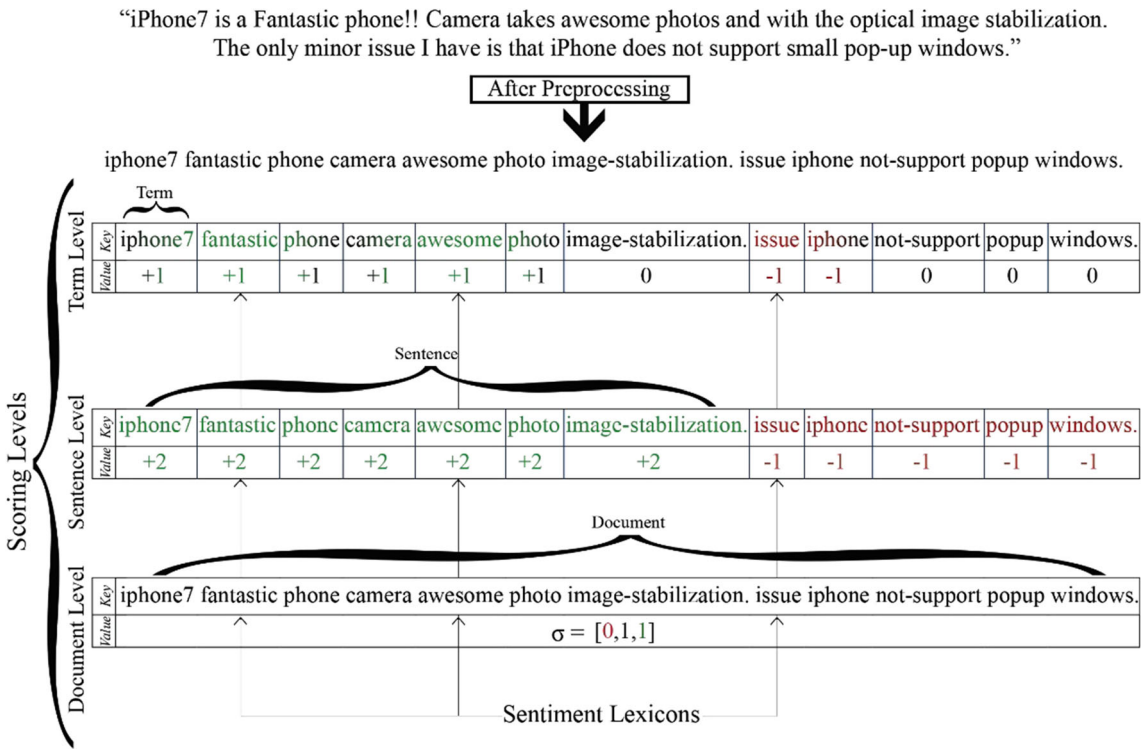

Fig. 1 Document, sentence and term levels of scoring in preprocessing phase

Algorithm 3: Document level of scoring

```
FOR each document d in D
  FOR each word w in d
    IF w in sentiment lexicons
      value(d) += score
    ENDIF
  ENDFOR
  FOR i=1 to S
    IF value(d)>min(i) and value(d)<max(i)
      sigma(i)=1
    ENDIF
  ENDFOR
  Emit (d, sigma)
ENDFOR
```

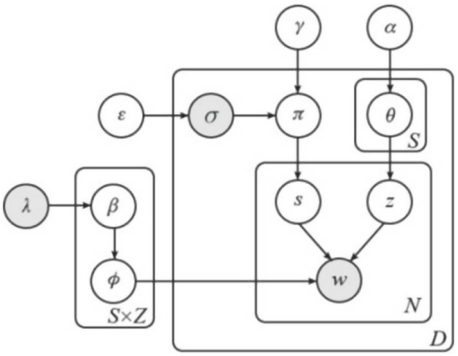

Fig. 3 Supervised Sentiment and Aspect graphical model

4.2 Supervised Sentiment and Aspect Model

The Supervised Sentiment and Aspect Model (SSAM) is a supervised topic model on an unlabeled corpus for classifying reviews by extending the unsupervised topic model LDA as shown in Fig. 2. SSAM considers document sentiment labels and term labels which are calculated in the preprocessing phase, during the generative process, where each document could have one or more sentiment labels. In contrast to most supervised topic models (Blei and McAuliffe 2010; Blei and Jordan 2003; Ramage et al. 2009), our proposed model not only considers document sentiment labels but also incorporates terms label to constrain sentiment–aspect word distribution prior for improving classification performance and extracting more discriminative aspects. Here both documents and terms are automatically annotated in the preprocessing phase by using a sentiment lexicon dataset. The graphical model of the proposed model is shown in Fig. 3. Let $C = \{d_1, d_2, \ldots, d_D\}$ be a set of documents; each document d be represented by a tuple consisting of a list of *(key, value)* pairs $d_i = \{(key_1, value_1), \ldots, (key_{N_d}, value_{N_d})\}$ and a list of binary sentiment presence/absence indicators $\sigma_d = (l_1, \ldots, l_S)$ where N_{d_i} is the length of document d_i and each *key* in *(key,value)* pair is a word member of vocabulary with V distinct terms $\{1, \ldots, V\}$ and *value* is the label of this word. Also let each $l_s \in \{0, 1\}$ and S is the number of sentiment labels. The formal definition of the generative process of SSAM is as follows:

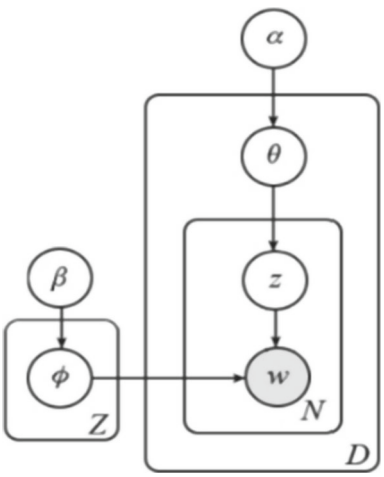

Fig. 2 Latent Dirichlet Allocation graphical model

Algorithm 4: The generative process of SSAM

```
For each sentiment s ∈ {1,...,S}
  For each aspect z = {1,...,Z}
    Compute β_{s,z} = λ_{s,z} × β
    Draw ϕ_{s,z} ~ Dir(β_{s,z})
For each document d ∈ {1,...,D}
  For each sentiment s ∈ {1,...,S}
    Draw θ_{d,s} ~ Dir(α_s)
    Generate σ_{d,s} ∈ {0,1} ~ Bernouli(ϵ_s)
  Compute γ_d = σ_d × γ
  Draw π_d ~ Dir(γ_d)
For each word w_i ∈ d
  Choose a sentiment label s_i ~ Mult(π_d)
  Choose an aspect z_i ~ Mult(θ_{d,s_i})
  Choose a word w_i ~ Mult(ϕ_{s_i,z_i})
```

The procedure for generating a word in document d under SSAM may be summarized in three steps. First one draws a sentiment label s from the per-document sentiment proportion π_d; in the next step, one chooses an aspect k from the per-document aspect distribution $\theta_{d,s}$ conditioned on the sampled sentiment label s. At the final step one chooses a word from the sentiment–aspect word distribution $\varphi_{s,z}$. The JST and ASUM models draw a multinomial mixture distribution π_d over all S sentiment labels, for each document d, from a Dirichlet prior γ. But we would like to restrict π_d to be defined only over the sentiments that correspond to its sentiment labels σ_d. Since the document-sentiment assignments s_i

(see line 12 in Algorithm 4) are drawn from this distribution, this restriction ensures that all the sentiment assignments are limited to the document's sentiment labels.

It is worth noting that if we use just the term level of scoring and set γ to a pre-defined constant (i.e., 0.1), then SSAM could be reduced to JST model. If we use the sentence level of scoring but do not incorporate the document's sentiment label, then SSAM could be like the ASUM model, and if we consider the term level of scoring with a pre-labeled corpus, our model works like feaLDA. Generative processes of JST, ASUM and feaLDA are different from the SSAM in that our proposed model incorporates learned supervised information in an effective way by introducing a transformation matrix λ and a document labels vector σ for encoding the knowledge achieved from the preprocessing phase to modify the Dirichlet priors of both sentiment–aspect word distributions and document specific sentiment distributions. SSAM exploits supervised information by using asymmetric priors γ and β. In the following, we discuss how to incorporate prior knowledge into the proposed model.

4.2.1 Incorporating document's sentiment labels

SSAM incorporates document's sentiment labels by introducing the document labels vector σ; to achieve this objective, we first generate the document's sentiment labels σ_d using a Bernoulli coin toss for each sentiment label s, with the sentiment labeling prior ε_s as shown in line 8 of SSAM generative process (Algorithm 4). We use the σ vector to restrict the document-sentiment Dirichlet prior $\gamma = (\gamma_1, \ldots, \gamma_S)$ as follows:

$$\gamma_d = \sigma_d \times \gamma \tag{1}$$

For example, suppose we have three sentiment labels, {*negative, neutral, positive*}, S = 3, and a document d has a vector of sentiments labels given by$\sigma_d = \{0, 1, 1\}$ then if π_d is drawn from a Dirichlet distribution with $\gamma_d = \sigma_d \times \gamma = (0, \gamma, \gamma)$ prior, this means the Dirichlet is restricted to sentiments *neutral* and *positive*. This fulfills our requirement that the document's sentiment labels are restricted to its own sentiment labels. The dependency of π on both γ and σ is indicated by directed edges from σ and γ to π in the plate notation in Fig. 3.

4.2.2 Incorporating terms or sentences label

Another type of supervised information considers term labels which are calculated from term and sentence levels of scoring in the preprocessing phase. In the existing supervised topic models, we usually set the Dirichlet prior of sentiment–aspect word distribution β to a symmetric value. Our experiments showed that incorporating term labels into the model could potentially improve the model classification performance. We encode the labeled terms into the SSAM model by introducing a word-sentiment association transformation matrix λ with dimension $V \times S$. For word w_i, its sentiment label association vector λ_{W_i} is calculated as follows:

$$\lambda_{w_i,s} = \frac{count(w_i \in sentiment\ s)}{\sum_{l=1}^{S} count(w_i \in sentiment\ l)} \tag{2}$$

$$\lambda_{w_i} = (\lambda_{w_i,s_1}, \ldots, \lambda_{w_i,s_S}) \tag{3}$$

Where the function *count()* enumerates all words w_i which are members of sentiment s, and also $\sum_{s=1}^{S} \lambda_{w_i,s} = 1$. For example, if there are three sentiment labels S = 3 and assume word *camera* with index w_t occurred 200 times in the sentiment label 1 and 80 times occurred in sentiment label 2 and 20 times occurred in sentiment label 3, has a corresponding association vector $\lambda_{w_i} = (200/300, 80/300, 20/300)$, we can then incorporate term labels into SSAM by setting

$$\beta_{w,s} = \lambda_{w,s} \times \beta \tag{4}$$

In this state, SSAM can ensure that a labeled term such as *"camera"* has a higher probability of being drawn from aspects associated with sentiment label 1. Initialization of β in SSAM is different from all other supervised and unsupervised topic models.

4.3 Learning and inference

From the SSAM graphical model shown in Fig. 3, the joint distribution of all variables (observed and hidden) can be factored into three terms:

$$\begin{aligned} P(w, z, s|\alpha, \beta, \gamma) &= P(s|\gamma)P(z|s, \alpha)P(w|s, z, \beta) \\ &= \int P(s|\Pi)P(\Pi|\gamma)d\Pi. \int P(z|s, \theta)P(\theta|\alpha)d\theta \\ &\quad \times \int P(w|s, z, \Phi)P(\Phi|\beta)d\Phi \end{aligned} \tag{5}$$

By integrating out π, θ and φ in the first, second and third terms on the RHS of Eq. (5), respectively, we obtain

$$P(s|\gamma) = \prod_d \frac{\Gamma\left(\sum_{k=1}^{S} \gamma_k\right)}{\prod_{k=1}^{S} \Gamma(\gamma_k)} \frac{\prod_s \Gamma(N_{d,k} + \gamma_k)}{\Gamma\left(N_d + \sum_{k=1}^{S} \gamma_k\right)} \tag{6}$$

$$\begin{aligned} P(z|s, \alpha) = \prod_d \prod_k \frac{\Gamma\left(\sum_{z=1}^{Z} \alpha_{k,z}\right)}{\prod_{z=1}^{Z} \Gamma(\alpha_{k,z})} \\ \times \frac{\prod_z \Gamma(N_{d,k,z} + \alpha_{k,z})}{\Gamma\left(N_{d,k} + \sum_{z=1}^{Z} \alpha_{k,z}\right)} \end{aligned} \tag{7}$$

Table 2 Meanings of the notations

D	The number of all reviews
V	The vocabulary size
Z	Number of aspects
S	Number of sentiments
z	Aspect
s	Sentiment
θ	Per-review sentiment–aspect distribution
φ	Sentiment–aspect word distribution
π	Per-review sentiment distribution
α	Dirichlet prior vector for θ
β	Dirichlet prior vector for φ
γ	Dirichlet prior vector for π
s_i	The sentiment of word i
z_i	The aspect of word i
s_{-i}	The sentiment assignments for all words except word i
z_{-i}	The sentiment assignments for all words except word i
w	The word list representation of review d
$N_{k,j,w}$	The number of times word w occurred in aspect j with sentiment label k
$N_{k,j}$	The number of words that are assigned sentiment k and aspect j
$N_{d,k,j}$	The number of words that are assigned sentiment label k and aspect j in review d
N_d	The total number of words in review d

$$P(w|s,z,\beta) = \prod_k \prod_z \frac{\Gamma(\sum_{v=1}^{V} \beta_{k,z,v})}{\prod_{v=1}^{V} \Gamma(\beta_{k,z,v})} \times \frac{\prod_v \Gamma(N_{k,z,v} + \beta_{k,z,v})}{\Gamma\left(N_{k,z} + \sum_{v=1}^{V} \beta_{k,z,v}\right)} \tag{8}$$

The notations are described in Table 2. In SSAM, we will assume that the documents and terms are multiply tagged in the preprocessing phase, at inference time. when the labels σ_d of the documents are observed, the document labeling prior ε is d-separated from the rest of the model given σ_d, and the sentiments per document prior γ_d is now restricted to the document d labels σ_d; therefore, we use collapsed Gibbs sampling (Griffiths and Steyvers 2004) to inference the latent variables θ, φ and π at each iteration of the markov chain. Sampling probability for choosing the sentiment and aspect of the ith word is given by

$$P(s_i = k, z_i = j | s_{-i}, z_{-i}, w, \alpha, \beta, \gamma) = \frac{N_{k,j,w_i}^{-i} + \beta_{k,j,w_i}}{N_{k,j}^{-i} + \sum_{i=1}^{V} \beta_{k,j,i}} \times \frac{N_{d,k,j}^{-i} + \alpha_{k,j}}{N_{d,k}^{-i} + \sum_{z=1}^{Z} \alpha_{k,z}} \times \frac{N_{d,k}^{-i} + \gamma_d}{N_d^{-i} + \sum_{s=1}^{S} \gamma_s} \tag{9}$$

Notations $N_{k,j,w}^{-i}$, $N_{k,j}^{-i}$, $N_{d,k,j}^{-i}$ and N_d^{-i} in this expression exclude the word i. Gibbs sampling (Algorithm 5) will sequentially sample each variable S and Z from the distributions over the observed variables of all other variables and data, until a stationary state of the markov chain has been reached. Then samples obtained from the markov chain are used to approximate the per-corpus sentiment–aspect word distribution

$$\phi_{k,j,w} = \frac{N_{k,j,w_i} + \beta_{k,j,w_i}}{N_{k,j} + \sum_{i=1}^{V} \beta_{k,j,i}} \tag{10}$$

per-document sentiment–aspect distribution

$$\theta_{d,k,j} = \frac{N_{d,k,j} + \alpha_{k,j}}{N_{d,k} + \sum_{z=1}^{Z} \alpha_{k,z}} \tag{11}$$

and per-document sentiment distribution

$$\pi_{d,k} = \frac{N_{d,k} + \gamma_d}{N_d + \sum_{s=1}^{S} \gamma_s} \tag{12}$$

Algorithm 5: Gibbs Sampling procedure of SSAM

Input: $\alpha, \beta, \gamma, Corpus$

Output: sentiment label and aspect assignment for all word tokens in the Corpus

Set $\beta_{w,s} = \lambda_{w,s} \times \beta$

Set $\gamma_d = \sigma_d \times \gamma$

Initialize π, θ, φ and all count variables

Initialize sentiment labels for all word tokens in the corpus using transformation matrix λ

Initialize aspects assignment randomly for all word tokens in the Corpus

FOR i=1 to max Gibbs Sampling Iterations do

  (S,Z, π, θ, φ**)= GibbsSampling(Corpus,** α, β, γ**)**

  FOR every 200 Gibbs Sampling Iterations do

    Update π, θ, φ **with new sampling results**

  Endfor

Endfor

4.4 Implementing SSAM on Spark framework

Spark (Zaharia et al. 2010) is a fast and general purpose engine for large-scale data processing framework which provides new features not previously available in Hadoop including caching, ease of use and many more. The detailed implementation of SSAM on Spark is shown in Algorithm 6. Here we first distribute data and parameters such as per-review sentiment distribution π and sentiment–aspect word distribution φ over P processors, with $\pi^p = \pi/p$ and $\varphi^p = \varphi$ on each processor, then collapsed Gibbs sampling procedure is executed on each processor, π^p and φ^p are

Table 3 Dataset statistics

Datasets	Amazon electronics	Amazon books
Number or reviews	143,828	38,473
Number of reviews with 3,4 and 5 stars	73%	77%
Average number of word/review+	102	67
Average number of word/review*	42	33
Corpus size+	15,822,742	3,064,464
Corpus size*	6,493,136	1,272,683
Vocabulary size+	470,779	172,669
Vocabulary size*	224,725	87,836

+ denotes before preprocessing and * denotes after preprocessing

updated locally at the same time; after the sampling, we calculate φ by collecting all locally updated $\widehat{\varphi}^{p}$ by using Eq. 13 then broadcast updated φ to all processors.

$$\varphi = \sum_{p} \widehat{\varphi}^{p} \tag{13}$$

Algorithm 6: Implementing SSAM on Spark

FOR i=1 to max Gibbs Sampling Iteration do
// Sampling on each processor
FOR each processor *P* in parallel do
Initialize φ^{p}, π^{p}
Sampling according to eq. 9.
Update φ^{p}, π^{p}
ENDFOR
Communication and synchronization
Collect φ^{p} **according eq. 13 and broadcast**
ENDFOR

5 Experimental setup

5.1 Dataset

In this paper, we use two different sets of Amazon reviews on electronic devices and books which we name *Electronics* and *Book*, respectively. These datasets are public on the internet.[1] We preprocessed the reviews by removing non-English alphabets and stop words based on a stop word list, stemming, extracting n-grams phrases and replace them in reviews. The final Book Dataset contains 38,473 documents, 87,836 unique words, and 1,272,683 word tokens in total; the Electronics Dataset contains 143,828 documents, 224,725 unique words, and 6,493,136 word tokens. Statistics before and after the preprocessing phase is summarized in Table 3.

[1] http://snap.stanford.edu.

In our experiments, two sentiment lexicons, namely MPQA[2] and appraisal lexicons,[3] were used to give a score to terms and documents in preprocessing phase.

5.2 Evaluation metrics

5.2.1 Sentiment classification accuracy

To specify the sentiment label of a review, we use the per-document sentiment distribution π (Eq. 12), such that a review is positive if the positive sentiment probability is equal to or a higher than negative sentiment probability, and is negative otherwise. For all datasets used here, each review is accompanied by a user rating on a scale of 1–5. Reviews rated as 1 or 2 stars are treated as negative and other ratings (3, 4 or 5) as positive.

5.2.2 Precision, recall and F-score

Average precision, recall, and F-score are used to evaluate the correctness of classified reviews in every sentiment label.

$$\text{precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \tag{14}$$

$$\text{recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{15}$$

$$f\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{16}$$

6 Experiments

In this section, we showed the experimental results of the SSAM model. We performed different experiments to evaluate our proposed model SSAM such as evaluating discovered sentiment–aspects by SSAM, presenting the sentiment classification performance of SSAM and comparing against two

[2] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon.

[3] http://opiniondetection.wikidot.com/resource.

Table 4 Such discovered sentiment–aspects by SSAM. (Labels are manually annotated)

Electronics				Books			
Picture quality (n)	Camera size (n)	Computer network (n)	Computer screen (p)	Romantic (p)	Politic (n)	Education (p)	War novels (n)
Noise	Camera	Internet	computer	Feel	Politic	Book	War
Picture quality	Battery	Network	Monitor	Heart	Culture	Course	Fear
Camera	Size	Issue	Bright	Love story	Middle east	Young	Soldier
Pixel	Kit	Wireless access point	Display	Classic	Democratic	High school student	Army
Resolution	Bulky	Plug	Screen	Friendship	Bad	Recommend	Force
Low quality	Camera size	Not work	Size	Leave	History	Collage	American
Contrast	Heavy	Bad	LCD screen	Romance	Inconsistent	Well write	Sadness
Amateur	Battery	Connect	Great	Love	Influence	Educate	Dark
Not clear	Camera bag	Slow	View	Interesting	Government	Child	Kill
Lens	Compact	DSL router	Inch	Life	People	Kid	Country
Distortion	Side	Home	Color	Emily	State	School	Happen
Color	Inch	Port	Sharp	Emotion	Republic	Parent	Human
Not good	Very small	File	New	Wonderful	Dissatisfaction	Teach	Action
Point	Not fit	Less	Video	Special	Foreign	Children	Critic
Low light	Pocket	Support	Show	Man	Policy	Think	Bad

weakly-supervised topic models ASUM and JST, comparing SSAM sentiment classification performance against two supervised methods Support Vector Machines (SVM) and Naive Bayes (NB) and finally comparing sentiment classification performance of SSAM in different levels of scoring. All the experiments reported here are averaged over 5 trials, and each trial randomly split the dataset into 80–20 for training and test. We ran SSAM with 1000 Gibbs sampling iterations. Note that the hyperparameters settings and sentiment lexicons are assigned similarly in all approaches.

6.1 Aspects discovery evaluation

In this experiment, discovered aspects coupled with a sentiment are evaluated. We use three criteria for extracted aspects: being coherent, being specific, and internal correlation. We applied SSAM on Electronics and Book review datasets and also evaluated the modeling power of SSAM based on the fore-mentioned three criteria. In this evaluation, we compared SSAM results with some other sentiment-topic models such as JST and ASUM. Here we analyze the extracted aspects under positive and negative sentiment labels. Some of the sentiment–aspects that SSAM discovered are presented in Table 4: aspects presented in Table 4 were generated in both positive and negative sentiment label each of which is shown by the top 15 aspect words. Inspecting the aspects extracted by SSAM, they are seen to be specific in every sentiment label, e.g., *camera size* is an aspect of camera which classified as a negative sentiment and the negative features such as *bulky*, *heavy* and *not fit* proved that. Another example of such extracted sentiment–aspects is politics where this aspect is classified as a negative sentiment because of existing negative sentiment words such as *inconsistent* and *dissatisfaction*.

Extracted aspects are coherent and informative in each class, e.g., the aspect *computer network* has a set of closely related and coherent words such as *internet*, connect, *DSL router*, *wireless access point* and also the aspect *picture quality* has words such as *low quality*, *not clear*, *contrast*, *resolution*. Another advantage of SSAM is the ability to extract multiword aspects and sentiments such as *picture quality*, *not clear*, *camera size*, *middle east*, *camera bag*, *lcd screen*, *not work*, *low quality*. Two hyperparameters, β and γ are tuned using incorporated supervised information. These two hyperparameters have a main role in extracting coherent aspects that are related to a specific sentiment.

6.2 Performance comparison of SSAM with two existing supervised methods

Our second experiment shows the classification results of SSAM on classifying a review as a positive sentiment or negative sentiment and also compares our model with two supervised methods, Naive Bayes (NB) and Support Vector Machines (SVMs). Beside the classification accuracy, three metrics Recall, Precision and F1 score are reported in Table 5. As will be seen from Table 5, SSAM outperforms NB by 13% in precision, 3% in F1 score and 6% in accuracy and also outperforms SVM by 5% in recall, 6% in precision and 6% in F1 score, but SVM outperforms both NB and SSAM in accuracy on Electronics Dataset. On the Books dataset SSAM outperforms NB by 24% in precision, 6% in F1 score

Table 5 Performance comparison of SSAM with two supervised approaches

Electronics				Books			
	Linear SVM	Naive Bayes	SSAM		Linear SVM	Naive Bayes	SSAM
Recall	85.02	**99.60**	90.45	Recall	74.12	**98.00**	81.17
Precision	84.11	77.84	**90.74**	Precision	92.03	68.35	**92.14**
F1-score	84.06	87.39	**90.61**	F1-score	84.16	80.53	**86.31**
Accuracy	**85.17**	77.61	83.90	Accuracy	73.74	69.07	**77.61**

Unit in % and numbers in bold face denote the best result in each metric

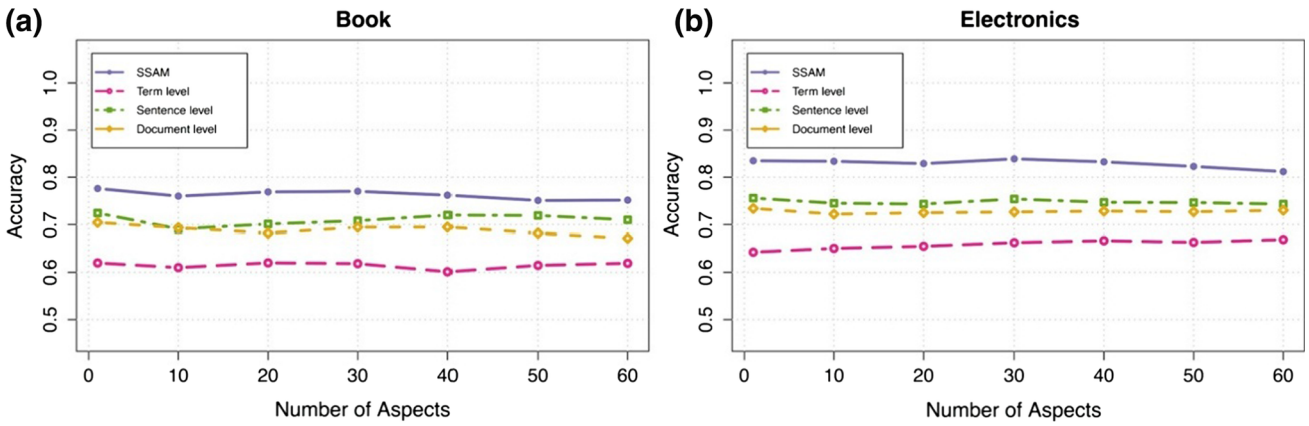

Fig. 4 Sentiment classification accuracy by the three different levels of scoring (Term, Sentence and Document) versus Different Aspect number settings **a** books dataset, **b** electronics dataset

Table 6 Performance comparison of SSAM with different levels of scoring

	Accuracy (%)			
	Scoring levels			
	Term	Sentence	Document	SSAM
Electronics	66.93	75.58	73.41	**83.90**
Books	62.02	72.43	70.44	**77.61**

Unit in % and numbers in bold face denote the best result in each metric

Table 7 Performance comparison of SSAM with two weakly-supervised sentiment-topic models

	Accuracy (%)		
	Topic modelling		
	ASUM	JST	SSAM
Electronics	78.83	69.94	**83.90**
Books	73.23	65.28	**77.61**

Numbers in bold face denote the best result in each metric

and 8% in accuracy, and also outperforms SVM by 7% in recall, 1% in precision, 2% in F1 score and 4% in accuracy. This demonstrates the effectiveness of SSAM in incorporating supervised information into the model inference. So applying a sentiment classifier such as SSAM that can offer a high precision and high recall to classify negative and positive sentiment while the majority of reviews are positive.

6.3 Performance comparison of SSAM with different levels of scoring

In this section, we show how the proposed model behaves with different aspect number settings on the above-mentioned datasets when different levels of supervised information (term level, sentence level, document level and mixtures of them) are incorporated. We present the sentiment classification accuracy of SSAM when incorporating different levels of supervised information, in Fig. 4. To achieve this objective, we conducted a set of experiments on SSAM by incorporating different levels of supervised information, with aspect number $Z \in \{1, 10, 20, 30, 40, 50, 60\}$. Table 6 shows the best classification accuracy results of SSAM by incorporating prior information extracted from the preprocessing phase at different levels. As can be seen from Fig. 4a, b, incorporating different levels of supervised information, i.e., term and document with multiple aspect settings on the Book and Electronics datasets, performs better than single level. Both Tables 6 and 7 show that at the term level of scoring, SSAM and JST have almost the same results and also at the sentence and document levels SSAM and ASUM have similar

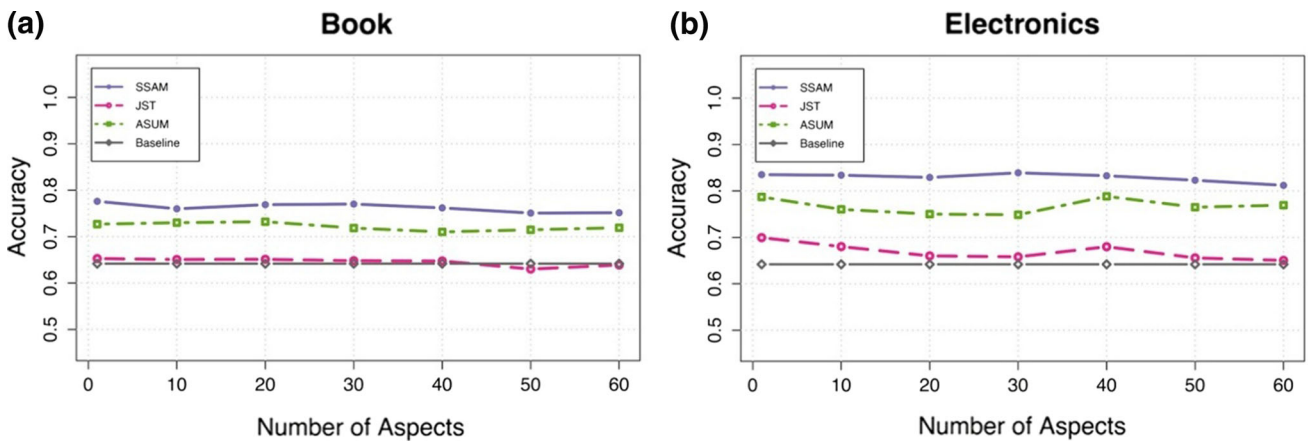

Fig. 5 Sentiment classification accuracy by the three topics models (SSAM, ASUM and JST) and baseline versus different aspect number settings **a** Books dataset, **b** electronics dataset

accuracy, but SSAM with both document and term levels of scoring gives a significant improvement over the others in all datasets.

6.4 Performance comparison of SSAM with existing weakly-supervised sentiment-topic modeling

In this experiment, we compare the sentiment classification performance of SSAM with other existing supervised or weakly-supervised sentiment-topic models (i.e., Aspect Unification Model ASUM and Joint Sentiment-Topic model JST): the sentiment classification accuracy results are presented in Figs. 5a and 4b and the best classification results are summarized in Table 7. In all aspect number settings, SSAM outperforms the other supervised and weakly-supervised sentiment-topic models. It can be seen from Table 7 that SSAM outperforms JST in accuracy by almost 14% and also outperforms ASUM by 5% on the Electronics dataset when the aspect number is set to $Z = 1$. The SSAM model outperforms JST by almost 11%. Although ASUM improves upon JST, it is worse than SSAM with its accuracy nearly 4% lower compared to SSAM on the Books dataset when setting aspect number to $Z = 30$. The baseline results in Fig. 5 are calculated based on the updated sentiment lexicon by counting the overlap of sentiment lexicon with each review in the corpus: if the count of positive sentiment words in a review is greater than the count of negative words, a review is classified as positive sentiment, and vice versa. As you can see, baseline results are below 65% for both datasets.

7 Conclusion

In this paper, we described a supervised sentiment aspect model (SSAM) which provides a novel framework for sentiments classification. While most of other supervised sentiment classification methods can only classify labeled reviews, SSAM is capable of incorporating different levels of supervision which are calculated in the preprocessing phase for improving sentiment classification performance. These supervised values are used to constrain the asymmetric Dirichlet prior of document-sentiment and sentiment–aspect word distributions. Results from different experiments show that SSAM outperforms two supervised models (i.e., SVM and NB) and also outperforms two weakly-supervised sentiment and topic models (i.e., JST and ASUM). Our proposed model only has a small sentiment lexicon dataset as supervised information in the preprocessing phase similarly to JST and ASUM. SSAM can extract implicit aspects, multiword aspects and multiword sentiments. Our proposed model used sentence structure and word order in the preprocessing phase and model inference.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Blei DM, Jordan MI (2003) Modeling annotated data. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informtion retrieval. ACM, pp 127–134

Blei DM, McAuliffe JD (2010) Supervised topic models. arXiv preprint arXiv:1003.0783

Church KW, Hanks P (1990) Word association norms, mutual information, and lexicography. Comput Linguist 16(1):22–29

David MB, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci 101(suppl 1):5228–5235

Ivan T, McDonald RT (2008) A joint model of text and aspect ratings for sentiment summarization. ACL 8:308–316

Jo Y, Oh AH (2011) Aspect and sentiment unification model for online review analysis. In: Proceedings of the fourth ACM international conference on Web search and data mining. ACM

Lacoste-Julien S, Sha F, Jordan MI (2009) DiscLDA: discriminative learning for dimensionality reduction and classification. In Advances in neural information processing systems, pp 897–904

Lim KW, Buntine W (2014) Twitter opinion topic model: extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In: Proceedings of the 23rd ACM international conference on information and knowledge management. ACM, pp 1319–1328

Lin C, He Y, Everson R, Ruger S (2012) Weakly supervised joint sentiment-topic detection from text. IEEE Trans Knowl Data Eng 24(6):1134–1145

Lin C, He Y, Pedrinaci C, Domingue J (2012) Feature LDA: a supervised topic model for automatic detection of web API documentations from the web. In: International semantic web conference. Springer, Berlin, pp 328–343

Liu Q, Gao Z, Liu B, Zhang Y (2015) Automated rule selection for aspect extraction in opinion mining. IJCAI 15:1291–1297

Mei Q, Ling X, Wondra M, Su H, Zhai CX (2007) Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp 171–180

Mimno DM, McCallum A (2012) Topic models conditioned on arbitrary features with dirichlet-multinomial regression. CoRR. arXiv:1206.3278

Poria S, Cambria E, Gelbukh A (2016) Aspect extraction for opinion mining with a deep convolutional neural network. Knowl Based Syst 108:42–49

Poria S, Chaturvedi I, Cambria E, Bisio F (2016) Sentic LDA: improving on LDA with semantic similarity for aspect-based sentiment analysis. In 2016 international joint conference on neural networks (IJCNN). IEEE, pp 4465–4473

Rahman MM, Wang H (2016) Hidden topic sentiment model. In: Proceedings of the 25th international conference on world wide web. International World Wide Web Conferences Steering Committee, pp 155–165

Ramage D, Hall D, Nallapati R, Manning CD (2009) Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 1. Association for Computational Linguistics, pp 248–256

Wang X, McCallum A (2006) Topics over time: a non-Markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 424–433

Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I (2010) Spark: cluster computing with working sets. In: Proceedings of the 2Nd USENIX conference on hot topics in cloud computing, HotCloud'10, Berkeley, CA, USA, 2010. USENIX Association, pp 10–10