EXPERT OPINION AND COHERENCE BASED TOPIC MODELING

Natchanon Suaysom, Weiqing Gu

Harvey Mudd College
nsuaysom@g.hmc.edu, gu@g.hmc.edu

ABSTRACT

In this paper, we propose a novel algorithm that rearrange the topic assignment results obtained from topic modeling algorithms, including NMF and LDA. The effectiveness of the algorithm is measured by how much the results conform to expert opinion, which is a data structure called TDAG that we defined to represent the probability that a pair of highly correlated words appear together. In order to make sure that the internal structure does not get changed too much from the rearrangement, coherence, which is a well known metric for measuring the effectiveness of topic modeling, is used to control the balance of the internal structure. We developed two ways to systematically obtain the expert opinion from data, depending on whether the data has relevant expert writing or not. The final algorithm which takes into account both coherence and expert opinion is presented. Finally we compare amount of adjustments needed to be done for each topic modeling method, NMF and LDA.

KEYWORDS

Topic modeling, expert opinion, Latent Dirichlet Allocation, Nonnegative Matrix Factorization, directed acyclic graph, tree, provable machine learning.

1. INTRODUCTION

In usual topic modeling, such as Latent Dirichlet Allocation (LDA) and Nonnegative Matrix Factorization (NMF), the algorithm would consist of preprocessing data, coming up with an appropriate cost function in certain metrics, using iterative solver to optimize certain cost function, and assigning the topics based on the results of certain solvers [1],[2].

However, the topic assignment part of the algorithm received a smaller attention from the research community [3]. This is because the assessment of how well the topic assignment being done is subjective. In order to address the subjectivity, we assume that there are certain public and experts opinion that believe certain words should belong in certain topics. After we come up with this idea, we search the literature for similar approach, the closest idea we can find is in sentiment learning [4].

However, the expert opinion is mostly used for calculating the accuracy of an algorithm rather than for improving topic assignments [4]. We create a novel approach which mimic how humans decide what should be in appropriate topics. We want the machine to adjust and balance the initial topics obtained from NMF and LDA automatically. In order to do so, we need to create certain data structures which would fit the purpose of this task. Tree and Directed Acyclic Graph (DAG) have received community attention to be the objects that capture opinion on words, [5], [4] [6]. In this paper we also created new data structure which combines Tree and DAG, we call the new structure TDAG, the reason we created such a data structure is we modeled expert opinions in

DOI: 10.5121/ijnlc.2018.7201

Tree or in DAG and sometimes both so we can integrate them. By doing so, we gain advantages in algorithmic sense that allows us to optimize a cost function, which captures the similarity between expert opinion and topic model results through each height of a tree, and allows the graph to have multiple roots. The advantage of this data structure is that it can capture data from table of content and from bag of words easily, which allows us to model both expert opinion from training data itself and from expert writing.

Assuming that the opinion on words are arbitrary, then we used the structure of TDAG to develop an algorithm that will make the clustering able to reflect expert opinion as well as possible, while still keeps the essential information on words that are not in expert opinion. For example, for NMF $||A - WH||$ is still small (so that it still makes a good clustering for words outside W_0).

To address topic modeling terminology and multiple new definitions on expert opinion, we started with the Background and Definition section. In Results section, we have two subsections to present both theoretical proof for further development and experiments on real world datasets. First we discuss theoretical results involving the algorithm design and we prove why the cost function reaches maximum value by our algorithm. Then in second part we present experimental results on datasets with and without expert opinion and an algorithm to obtain expert opinion (in case no expert opinion writing is available in the area). Lastly, we give our conclusions and offer suggestions for future work.

In this paper, we first discuss the background of the topic modeling method, define expert opinion and cost function in Background and Definition. In Results section, we have two subsections. First we discuss theoretical results involving the algorithm design and we prove why the cost function reaches maximum value by our algorithm. Then in second part we present experimental results on large datasets and an algorithm to obtain expert opinion (in case no expert exists in the area).

2. RELATED WORK

We investigated relevant past papers on the evaluation and adjustment of topic assignment in topic modeling. First, [7] studies the interpretation of humans on topic modeling results. The works addresses the subjective evaluation of human by using word intrusion idea, which measures how semantically cohesive topic from the model is, and makes the evaluation more accurate. This requires human involvement in the judgment, so we instead automatically obtained expert opinion to make the evaluation more formal. Other interpretation using different metrics are possible. [8] uses metric in stability analysis to discuss the evaluation of the number of topics that should be used in topic modeling. Although this metric is very accurate and perfectly suits our purpose, it requires a creation of $n \times n$ matrix for dataset of size n, so we cannot use this method for our Twitter dataset with 4 millions tweet because the matrix would be too large to store. [9] explores the advantage of each topic model using a metric called *coherence*, which takes into account the pattern of words inside and outside each topic, and is a general metric that works with all topic modeling method that we use to evaluate our expert opinion based model on large data.

Next we explored the idea of integrating expert opinion into topic modeling. [5] takes advantage of large data available on the internet to integrate opinion to a topic using modified PLSA. This is a direct integration to the existing topics from topic models. To algorithmically design expert opinion, we investigated the existing data structure used for representing expert opinion, and found the use of both tree [4] and directed acyclic graph [6] [10]. We found great advantage for both data structure and decided to combine them into new data structure, TDAG.

Finally we investigated the idea of topic rearrangement to make the topic assignment more accurate. [3] mainly studies topic assignment on NMF and uses game theory and game dynamics to adjust the topics with the method Game Theoretic NMF (GTNMF), and test the algorithm on

different datasets. This is what we expect to improve by allowing the approach to a more general topic modeling methods.

3. BACKGROUND AND DEFINITION

3.1. Topic Assignment

Let T be the set of all documents in the datasets that has set of words W. Let $n = |T|$ and $m = |W|$. The general approach to do clustering is to obtain a bag of words represented in matrix $A_{n\times m}$ where $A_{i,j}$ denote the number of times document with index i contains word with index j. Then topic modeling techniques such as LDA and NMF are used to reduce data into two matrices $W_{n\times k}$ and $H_{k\times m}$, where k is the number of topics, which is a parameter we can pick. H represents the relationship between words and topics, so that H_{ij} is the likelihood that topic indexed by i contains word indexed by j. W generally represents the relationship of documents to topics where W_{ij} is the likelihood that document indexed by i is in topic indexed by j. In order to decide which topic each document $t \in T$ belongs to, we consider the maximum likelihood estimate, which is the following formula

$$\text{Topic}(t) = \arg\max_{1\leq j\leq k} W_{I(t),j} \tag{1}$$

which is the column index of W that contains highest probability for document t to be in that topic. This is usually the final step of topic modeling and is the step that we focus on in this paper. Although NMF and LDA themselves are carefully studied and there are multiple algorithms to solve them, this final step of assigning topics receives less attention from the research community [3]. The assessment of this step is usually done by considering the top words (words with high frequency) of each topic and decided by eyeballing to see if they naturally make sense. This makes the improvement of this final step subjective. In order to assess whether this is done properly or not and remove subjectivity, a central expert opinion that is not biased has to be used. Since the evaluation concerns how likely certain words should belong together, an expert opinion should reflect that.

Although tree is popular for representing relationship between words, we expect the structure to have multiple roots and be easily traversed, and these two properties are well known in DAG (Directed Acyclic Graph).

In our applications, these two structures need to be integrated into a new data structure which we called TDAG and will be defined in the next section. As we can see from the following Figure 1 and 2 that each subtree in TDAG will contain its own root and each children to be words that are related to their parents. The need to have multiple roots is clear in this sense, because it is impossible to find only a word that governs all words in the documents. The number of roots and where they are can be arbitrarily decided by an expert. As we can see, in order for us to effectively represent expert opinion, a data structure combination of DAG and tree is needed. The advantage is that we can traverse the DAG to optimize topic assignment based on each word, and this can be done quickly because each word is organized into its corresponding level of the tree.

3.2. Data Structure

Definition 1 A graph TDAG is a graph $G : G(h, f)$ with $V(G) = \{v_{i,j}, 1 \leq i \leq h, 1 \leq j \leq f(i)\}$, where h is the height of TDAG and $f(i)$, for $1 \leq i \leq h$, is the number of vertices of height i (so that $v_{i,j}$ is the j'th vertex in height i). Furthermore it has the properties

1. Tree Property : For each vertex $v_{i,j}$ where $i \geq 2$, there exists at least one vertex $v_{i-1,j'}$ in height $i-1$ such that $v_{i-1,j'} \to v_{i,j} \in E(G)$, (all such vertices are called *parents* of $v_{i,j}$ and is denoted by $P(v_{i,j})$).

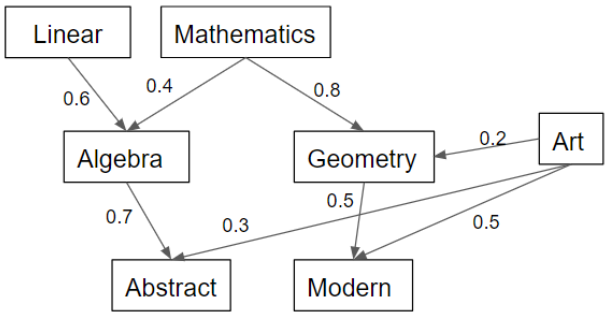

Figure 1: Example of expert opinion in academic topics in Mathematics and Art as TDAG with normalized likelihood. The probability of each node's parents sum to 1.

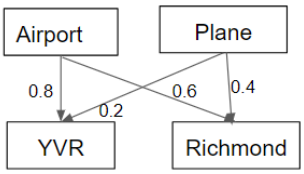

Figure 2: Example of Expert Opinion in Twitter Data. Here YVR is the code for Vancouver International Airport which is in Richmond, British Columbia, Canada.

2. DAG Property : The TDAG is topologically sort, which means that the edge cannot point back to vertices with less height, or that there is no edge of the form $v_{i,j} \to v_{i',j'}$ where $i > i'$

Definition 2 Let W be the set of all words in the document, then an *expert opinion* is a TDAG G : $G(W_0)$ with vertices $W_0 \subseteq W$. Furthermore, each directed edge $e = v_{i,j} \to v_{i+1,j'}$ has a positive weight $p(e)$ that shows the expert's estimation of the probability that the words $v_{i+1,j}$ belong to the topic containing $v_{i,j}$ and $v_{i+1,j'}$. Note that this automatically requires that $\sum_{v' \in P(v)} p(v' \to v) = 1$. (Otherwise the weight in each edge has to be normalized to sum to 1)

3.3. Expert Opinion

Note that if $W_0 = W$, or equivalently the expert has opinion on all words in the datasets, then we have enough information to categorize topic without doing any topic modeling by just picking the best roots that each document belongs to. This is unlikely to happen because in general the data contains so much noise that the expert cannot address them all, or equivalently, $|W_0| << |W|$, which means that expert only has opinion on certain topics. This leads us to define a cost function that tells us how well the topic assignment reflects expert opinion and how well the topic assignment reflects the result from NMF or LDA for words outside expert opinion.

3.4. Cost Function

First we define a cost function that captures how much the topic assignment reflects the expert opinion. The cost function should calculate the expected value that pair of words in the same topic is in expert opinion. This leads to the following definition.

Definition 3 Let $0 < k < 1$ be a regularization exponent. For each word w define $P(w)$ to be all

parents of w in tree W_0, and $f(w,k) = \sum_{w' \in P(w)} (T_t(w,w'))^k p(w,w'))^{1-k}$ where $T_t(w,w')$ is the number of documents containing w in topic containing w, w'. Define the cost function

$$C(W_0, k) = \sum_{w \in W_0} \frac{f(w,k)}{N(w)^k}, \tag{2}$$

where $N(w)$ is the number of documents in T containing w.

The quantity, when $k = \frac{1}{2}$ calculate $\mathbb{E}[\sqrt{X(w,w')}]$ where $X(w,w')$ is the number of time pair of words w, w' appear in a topic. This is restrictive, so the regularization constant $0 < k < 1$ can be used to emphasize the expert opinion or the internal topic structure. When $k \to 0$, we give full control to the probability in expert opinion, and vice versa when $k \to 1$.

We found that when $k = \frac{1}{2}$, the algorithm we used could reach the proven maximum. For the cost functions to reflect the results for NMF, we can just truncate the rows of A and columns of H that has expert opinion. Suppose the resulting matrices are A',H', then it can be seen that $\|A - WH\| \approx \|A' - W'H\|$, this means that we can still use the same cost function for NMF.

Recall that a topic coherence metric is defined in [9], as follow

$$\text{Coherence}(T) = \sum_{(w_i, w_j) \in W(T) \times W(T)} \text{Score}(w_i, w_j, \epsilon)$$

where ϵ is a smoothing factor, and the two scoring method we studied are the following

$$\text{UCI}(w_i, w_j) = \log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)}$$

where $p(w_i, w_j)$ is the co-occurence frequencies of external corpus such as our training set, and another scoring metric is

$$\text{UMASS}(w_i, w_j) = \log \frac{D(w_i, w_j) + \epsilon}{D(w_i)}$$

where $D(\mathbf{w})$ is the number of times **w** occurs in a document.

Topic coherence metric is used to see how well a certain topic modeling techniques divides documents into topics.

4. RESULTS

4.1. Theoretical Results

In order to adjust topic assignment based on the cost function, we design an algorithm which traverse TDAG efficiently and rearrange the documents balancely according to the expert opinion. It is also important for us to make sure that the algorithm does not interfere the internal structure of the underlying topic model. For example, the rearrangement does not change much of the value of $\|A - WH\|$ for the rest of the datasets. The following algorithm we design, EOB NMF/LDA (Expert opinion based NMF/LDA) also uses randomness to make sure the documents are not picked based on certain indexing, and to show that our algorithm is robust.

Algorithm 1 Expert opinion based (EOB) NMF/LDA

Require: Set of documents T, number of topics k, expert opinion $W_0 \subseteq W$

for i=1:h **do**
  Do NMF/LDA Analysis on T to obtain matrices $W_{document \times topic}$.
  Obtain topic for each document as discussed earlier using

$$\text{Topic}(t) = \arg\max_{1 \leq j \leq k} W_{I(t),j}$$

  for w_1 in $T(W_0, i)$ **do**
    for w_2 in $T(W_0, i+1)$ **do**
      if w_1 and w_2 is in top words of topics t **then**
        Move all documents with word w_1 to the subtopic based on their probability. This means that if there are N documents with word w_1, randomly picked $Np(w_1, w_2)$ of them and move them to topic t. (If a document satisfies this for different pairs of (w_1, w_2), Theorem 3 can be used as one way to decide which topic it should be moved to)
      end if
    end for
  end for
end for

Ensure: Set of topics with minimum cost function.

Next we prove theoretical results based on Algorithm 1. With certain conditions, we prove that the maximum value of cost function is reached. We also show in later sections that in real datasets even when the condition is not met, the difference between expected maximum and cost function is small, and therefore we still can use the method.

Theorem 1 Let G be the graph from expert opinion on $W_0 \in W$. Assume a condition that no three words that are adjacent in G from expert opinion appear in any document, then $C(W_0, \frac{1}{2})$ reaches a maximum and minimum, and the maximum is achieved when the amount of documents in topics are proportional to each other based on their probabilities in W_0, and the minimum is acheived when all assignments in W_0 are wrong.

Proof

1. To prove that the maximum is achievable, note that from Cauchy-Schwarz inequality

$$f(w, \frac{1}{2}) \leq \sqrt{\sum_{w' \in P(w)} T_t(w, w') \sum_{w' \in P(w)} p(w, w')}$$

   By definition of weighted tree the sum of probabilities in the parents will sum to 1. Which means that

$$\sum_{w' \in P(w)} p(w, w') = 1$$

   Since no document contains three words that are adjacent in G, it is not possible for a document t to be calculated in both topics w, w'_i and w, w'_j, so that any two topics $T_t(w, w_i)$, $T_t(w, w_j)$ are disjoint. This means that $\sum_{w' \in P(w)} T_t(w, w')$ must count at most the number of all document containing word w, which is $N(w)$, so we have that

$$\sum_{w' \in P(w)} T_t(w, w') \leq N(w)$$

   Putting both equations together we have that,

$$C(W_0, \frac{1}{2}) \leq \sum_{w \in W_0} \frac{f(w)}{\sqrt{N(w)}} \leq \sum_{w \in W_0} \frac{\sqrt{N(w)}}{\sqrt{N(w)}} = |W_0|.$$

And the equality holds from Cauchy-Schwarz's equality condition, which is when $\frac{T(w,w')}{p(w,w')}$ are constants for all w', or equivalently when the amount of moved documents is proportional to their probability, which is exactly what our algorithm does. This means that our algorithm allows this cost function $C(W_0, \frac{1}{2})$ to reach maximum value after running on all heights of the TDAG.

2. For the minimum value, when the document containing w is never put on a topic containing w, w', the value $T_t(w, w') = 0$ for all w and its parent w'. This means that the cost function $C(W_0, k) = 0$ is possible.

Theorem 2 When $k \neq \frac{1}{2}$, $C(W_0, k)$ has a maximum value, if the dataset satisfies condition presented in Theorem 1 , the maximum value of the cost function is also acheived by Algorithm 1.

Proof We can use Hölder's inequality. Let $p = \frac{1}{k}$ and $q = \frac{1}{1-k}$, then we can see that $\frac{1}{p} + \frac{1}{q} = 1$, so we get that

$$f(W_0, k) = \sum_{w' \in P(w)} T_t(w, w')^{\frac{1}{p}} \sum_{w' \in P(w)} p(w, w')^{\frac{1}{q}}$$

$$\leq (\sum_{w' \in P(w)} T_t(w, w'))^{\frac{1}{p}} (\sum_{w' \in P(w)} p(w, w'))^{\frac{1}{q}}$$

using

$$\sum_{w' \in P(w)} T_t(w, w') \leq N(w)$$

we get

$$f(W_0, k) \leq N(w)^{\frac{1}{p}} = N(w)^k$$

so that

$$C(W_0, k) \leq \sum_{w \in W_0} \frac{N(w)^k}{N(w)^k} \leq |W_0|.$$

The maximum value is reached when there is a constant α such that

$$\alpha(T_t(w, w')^{\frac{1}{p}})^p = ((p(w, w')^{\frac{1}{q}})^q$$

which means that $T_t(w, w')$ and $p(w, w')$ are proportional to each other, and that is what the algorithm does.

In general, however, the condition in Theorem 1 might not be met. In this case we will need to pick only one topic from multiple choices of topics to move the document. We resolve this problem by using the following theorem.

Theorem 3 Suppose that a document t contains words $w_1, w_2, \ldots, w_k$ that appears in TDAG, then if we use $k = \frac{1}{2}$ and decide to move the document based on the word w_i with maximum value of

$$\frac{p(w_i, w')}{N(w_i) T_t(w_i, w')}$$

where w' is any parent of w_i, then we get the maximum cost function value.

Proof We consider the quantity

$$\Delta = \frac{\sqrt{p(w_i, w')}}{\sqrt{N(w)}}(\sqrt{(T_t(w_i, w') + 1)} - \sqrt{T_t(w_i, w')})$$

which is the value of cost function we gain from putting a document t based on edge w_i, w'. We can write

$$\Delta = \frac{1}{\sqrt{T_t(w_i, w') + 1} + \sqrt{T_t(w_i, w')}} \frac{\sqrt{p(w_i, w')}}{\sqrt{N(w)}}$$

$$\approx \frac{1}{2}\sqrt{\frac{p(w_i, w')}{N(w_i)T_t(w_i, w')}}$$

since we want to maximize Δ across all possible edges in TDAG, the quantity gives the best increase in cost function by choosing maximum edge with the quantity, as desired.

Lastly, we want to make sure that our algorithm does not change too much of the internal structure of the clustering. In order to assess how well the algorithm works with words that are not in expert opinion, some measures that have been used are such as **Topic Coherence** metric [9]. A few versions of that has been defined in our Backgrond and Definition section. Note also that this topic coherence metric has been used for both NMF and LDA, so we can use them for analyzing our algorithm for both topic modeling techniques [9]. We only focus on UMASS coherence with $\epsilon = 1$ in this case.

Algorithm 2 EOB NMF/LDA Adjusted for UMASS Coherence

Require: Set of documents T, number of topics k, expert opinion $W_0 \subseteq W$

for i=1:h **do**
  Do NMF/LDA Analysis on T to obtain W and assign topics to documents as discussed in Algorithm 1
  for t_1 in $T(W_0, i)$ **do**
    if t_1 is in some topic k **then**
      Let $S(t_1)$ be the set of documents with word t_1.
      Let $A(t_1) = \{T : (t, t') \in W(T), t' \in P(t_1)\}$. For each $l \in A(t_1)$ found from edge (t, t') let $p(l)$ be its probability $p(t, t')$.
      Compute $R(t_1) = \{(t, \text{argmax}_{T \in A(t_1)})|t \cap W(T)|) : t \in S(t_1)\}$
      For each (t, T) in $R(t_1)$, if $|t \cap Top(T)| \geq 2$ move t to topic T. Set $S(t_1) := S(t_1) \backslash t$.
      For each $l \in A(t_1)$, $p(l) := p(l) - \frac{|R(t_1,l)|}{|S(t_1,l)|}$.
      Move the rest of documents in $S(t_1)$ to the topic based on their reduced probability $p(l)$.
    end if
  end for
end for

Ensure: Set of topics with minimum cost function.

We are going to explain the terminology in Algorithm 2 as follow. The set $A(t_1)$ is the set of all parents of t_1, and $R(t)$ computes the set of documents with highest intersection in the set of the top words. Since those t have high intersection it is likely that they are in topic T, so we move them. The probability $p(l)$ is then adjust to counterbalance this change as follow. We can see that for each t_1, the proportion of moved document for each of its parent is $p(l) - \frac{|R(t_1,l)|}{|S(t_1,l)|} + \frac{|R(t_1,l)|}{|S(t_1,l)|}$ which

Topic	10 Top Words in NIPS datasets after LDA
1	'connection', 'input', 'vectors', 'bit', 'feature', 'connections', 'distributed', 'vector', 'recognition', 'data', 'best', 'current', 'features', 'analog', 'output'
2	'kernel', 'kernels', 'space', 'learning', 'svm', 'support', 'machines', 'feature', 'machine', 'product', 'spaces', 'approach', 'functions', 'use', 'section'
3	'line', 'motion', 'data', 'analog', 'field', 'network', 'image', 'fig', 'processes', 'process', 'networks', 'term', 'circuit', 'problem', 'solution'
4	'pca', 'principal', 'analysis', 'component', 'data', 'components', 'subspace', 'linear', 'variance', 'algorithm', 'nonlinear', 'eigenvalues', 'dimensional', 'reduction', 'covariance'
5	'greedy', 'functions', 'submodular', 'sets', 'function', 'elements', 'gain', 'extension', 'element', 'problem', 'class', 'hard', 'selected', 'ground', 'near'

Table 1: Example of top words in topics of the NIPS datasets after LDA.

gives $p(l)$, that means that the amount of moved document is proportional to its weight probability, thus giving maximum value of cost function. On the other hand, the coherence also increases because more documents have higher intersection with its top words. Suppose the algorithm changes the assignment from T to T' so that one pair of words appears one more time in a topic, then

$$e^{C(T')} - e^{C(T)} = \frac{D(w_i, w_j) + 1 + \epsilon}{D(w_i) + 1} - \frac{D(w_i, w_j) + \epsilon}{D(w_i)}$$

$$= \frac{D(w_i) + 1 - D(w_i, w_j)}{D(w_i)(D(w_i) + 1)} > 0$$

because clearly $D(w_i) \geq D(w_i, w_j)$. So that $C(T') > C(T)$. This means that at any step of the algorithm, the coherence always increases, so that our algorithm always increase topic coherence while maximizing the cost function value.

5. EXPERIMENTAL RESULTS

5.1. Dataset With Expert Opinion

A dataset with expert opinion we consider is the dataset of *NIPS Conference Papers 1987-2015 Data Set* [11] which contains 5861 papers and 11463 words in the bag of words matrix. Topic modeling techniques such as LDA can detect topics in this dataset, but we can improve the results with our techniques. The expert opinion we used is the table of content from [12], and is processed by Algorithm 5.

For the following algorithm, the depth of subchapter is the size of the numbers in its description. For example, the section number 1.2.3 is of depth 3 and section number 1.2, and section 1.2.3 would be a *children* of section 1.2.

This algorithm uses expert opinion from all chapters to adjust the topic assignments, and the result of cost function after using some chapters is shown below. In general, the cost function for all chapters increase after the algorithm as shown in Figure 5.

5.2. Dataset Without Expert Opinion

Some noisy datasets may not have expert opinion, and we pick Twitter dataset to satisfy such criteria. Our dataset contains 4 Million tweets in Vancouver. The text file contains over 700 MB and about 40 Million words. Each tweet has the property that it contains less than 144 characters, so it contains, on average, less than 20 words, which makes topic assignment difficult due to

Algorithm 3 Obtain Expert Opinion from Table of Contents

Require: Table of contents (TOC) T, Chapter number C
1: Calculate the height of T for each chapter
2: **for** c in $1 : C$ **do**
3: **for** h in $1 : H$ **do**
4: Create a TDAG where words in chapter name with depth h has an edge to that of its children with depth $h + 1$ with equal weights.
5: We obtain the expert opinion W_0^c
6: **end for**
7: **end for**
8: Use Algorithm 2 on the expert opinion W_0
Ensure: Set of topics assignments with minimum cost function.

Topic	Tweets in the Dataset
1	I'm at Steve Nash Sports Club (Vancouver, BC))
2	"See ya later #Vancouver you've been great! But now in the words of #WillSmith it's welcome to #Miami"
3	An economy based on endless growth is unsustainable *drop the bass*
4	Collaborative Initiative this morning with Project Management. Social media marketing magic is about to happen.
5	Happy Monday how was everyone's weekend? Let's do this!

Table 2: Example of tweets in the Vancouver dataset

noises. The sample data is shown in Table 2. The dataset, then, is preprocessed to only contains useful data and put into bag of words matrix by Algorithm 4.

Algorithm 4 Preprocessing

Require: Set of tweets T
Ensure: A bag of words matrix A
1: Obtain all the words that appear in the tweets.
2: Remove all words not in the top words list and words that appear altogether less than 10 times. Suppose that the list of words has size w.
3: Create an n-by-w word count matrix A, where the entry A_{ij} is the number of times that tweet i contains the word j
4: Obtain matrix A representing a bag of words.

We can see from Table 3 and 4 that the topic results from NMF and LDA that talks roughly about

Topic	10 Top Words in Tweets in the Dataset after NMF
1	'im', '604insomnia', 'st', 'ave', 'sure', 'gonna', 'bus', 'westminster', 'sorry', 'glad'
2	'vancouver', 'downtown', 'aquarium', 'vancity', 'bc', 'city', 'north', 'gastown', 'sunset', 'yaletown'
3	'yvr', 'airport', 'international', 'yvrairport', 'richmond', 'flight', 'vancouver', 'village', 'waiting', 'bye'
4	'british', 'columbia', 'burnaby', 'richmond', 'ubc', 'university', 'bcit', 'station', 'skytrain', 'vancouver'
5	'game', 'watching', 'hockey', 'watch', 'vancanucks', 'win', 'play', 'whitecapsfc', 'habs', 'team'

Table 3: Example of top words in topic of tweets in the Vancouver dataset after NMF

Topic	10 Top Words in Tweets in the Dataset after LDA
1	'new', 'way', 'bar', 'christmas', 'lets', 'keep', 'making', 'westminster', 'brunch', 'shopping'
2	'downtown', 'move','cactus', 'foursquare', 'forward', 'gate', 'forget', 'gastown', 'garden', 'games'
3	'send', 'photos', 'tried', 'yvrairport', 'funny', 'full', 'fun', 'gallery', 'future', 'friends'
4	'im', 'vancouver', 'canada', 'british', 'columbia', 'market', 'sunset', 'happening', 'bcplace', 'caf'
5	'one', 'hockey', 'community', 'course', 'least', 'side', 'fans', 'todays', 'team', 'smile'

Table 4: Example of top words in topic of tweets in the Vancouver dataset after LDA. Note some of similarity between each corresponding topics to that of NMF.

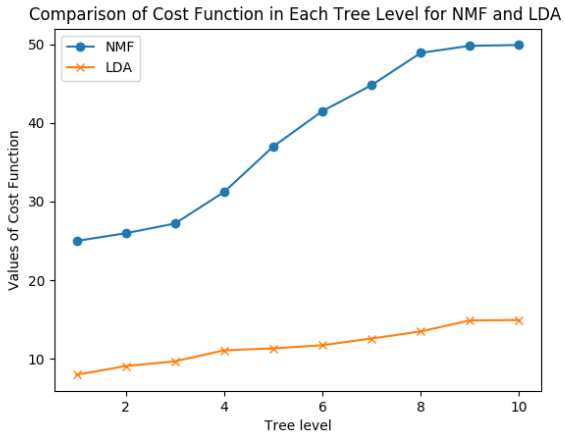

Figure 3: Cost Function Value Comparison with NMF and LDA with regularization parameter $k = \frac{1}{2}$

the same topic have different top words. The discussion about this differences appear in [9]. We can see from the results of top words of topics that, it is possible that some tweets can be in the wrong topic. For example, a tweet about airport belongs to a topic about Vancouver downtown, then it should be moved to the appropriate topica about airport. However, it is likely that there are many such correct topics, so that Algorithm 1 needs to be used to distribute correctly to maximize the cost function shown in Theorem 1.

Note that this dataset does not necessary have an expert opinion, so we need to derive them from training data. First we derive how we obtain the TDAG W_0 from each dataset T. Suppose the datasets is divided into T_{train} and T_{test}, then we try to learn useful TDAG W_0 from $W_{T_{train}}$ as shown in Algorithm 5. The algorithm derives the TDAG based on the ratio of the number of times certain words we pick appear in the training set, then adjust the TDAG into appropriate height and number of vertices and edges that reflect the relationship between top words.

The following graph shows the experiment of cost function with $N = 250$, $h = 10$ and 100 topics in Figure 3. The change in coherence is shown in Figure 4. Note that the cost function is very close to its expected maximum, but since the condition in Theorem 1 is not met, the cost function differs slightly from the expected maximum, but the result is acceptable.

Different value of regularization parameter k are also picked and shown on Figure **??**. Although we expect the maximum to be identical for all k if the condition is met, this is not always the case for real data. Initially as one can see in Figure **??** the cost function values are almost the same, but

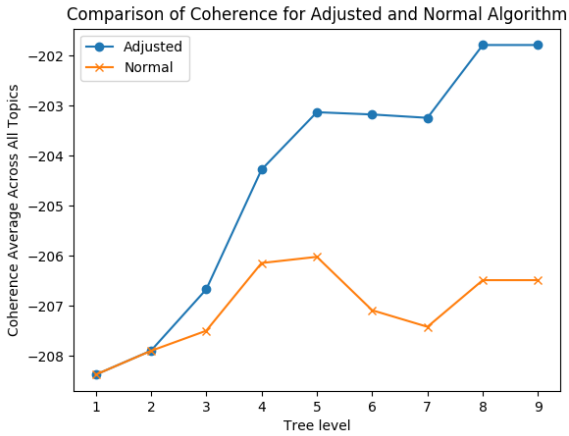

Figure 4: Comparison Between Average UMASS Coherence Across all Topics in Adjusted Algorithm 2 and Normal Algorithm 1 using NMF

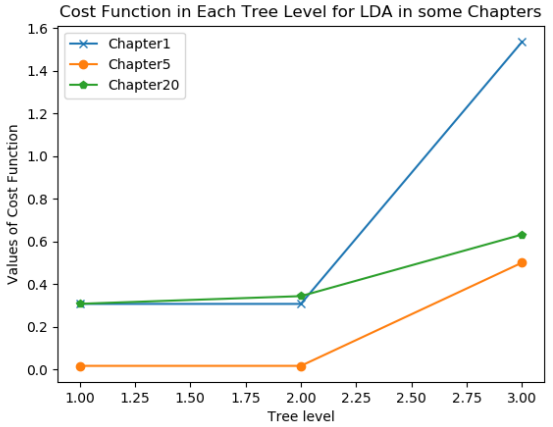

Figure 5: Cost Function in Each Tree Level for LDA in some Chapters

Algorithm 5 Obtain Expert Opinion from Training Set

Require: Set of tweets T, number of words N, tree height h.

1: Find W, the set of all words appear in T.
2: Find the N words that appear with most frequency in W, let them be W_N.
3: **for** w_1 in W **do**
4: **for** w_2 in W **do**
5: Find the number of tweets containing w_1 and w_2, denote them $N(w_1, w_2)$.
6: **end for**
7: **end for**
8: Define $p(w_1, w_2)$, the probability in the graph G to be $\frac{N(w_1,w_2)}{\sum_{w\in W} N(w_1,w)}$.
9: Let the words in level i be the top $\frac{Ni}{h}$ to $\frac{N(i+1)}{h}$ words.
10: Define the graph G to have the node to be words in W, and the node in each level and the weight defined above.

Ensure: G, an expert opinion TDAG learned from training set.

as we go down each level of TDAG, the value of cost function starts to differ since the condition is less likely to be met. The lower k means we expect the expert opinion to have more impact, and the user of the algorithm can set appropriate cutoff to stop the algorithm when desired value is achieved.

6. DISCUSSION AND CONCLUSION

Our Algorithm 2 improves the topic assignment done in Equation (1) by allowing experts opinion in the form of TDAG to adjust the assignment based on the cost function in Equation (2). This novel algorithm provides a way for a machine to adjust the topic assignment automatically. This is possible because we provided theoretical results. For example, we proved in Theorem 1 and 2 that the maximum value of cost function is achievable, and when $k = \frac{1}{2}$ the maximum value is nearly reached, allowing users to set cutoff to stop at partial point of a TDAG or run the whole algorithm through the whole TDAG for maximum value. In the field of datasets where expert opinion has strong influence toward contents in the documents, Algorithm 2 can be applied directly, and if the condition is almost satisfied, the cost function will reach a value closed to the maximum value. Otherwise expert opinion can be learned from the training set as shown in Algorithm 5, where probabilities are calculated by the ratio of words that appear in training sets. Our results show that in both cases, the algorithm work well on large and general datasets.

It is worth nothing that this approach is particularly useful on improving text-based prediction such as Twitter location prediction done in [13],[14]. Text-based prediction is done using topic modeling on training sets with interested properties to predict. And then project the results onto the testing sets to get topics assignment that predicted the properties. Since topic assignment is often done using argmax method such as in Equation (1), and the data is modified multiple times throughout the algorithm, the resulting topic assignment usually needs to be improved. Because the expert opinion can be learned from the given training sets, such improvement can be done directly by applying our Algorithm 2 to maximize the expectation and minimize the changes to the coherence. In addition our method leads to more accurate predictions on documents that contain expert opinion.

We expect that this approach can also be applicable on other type of text-based analysis. One such application that we are exploring is to do combinations of different topic modeling techniques. A stretch goal would be using expert opinion on other machine learning techniques that may or may not be related to text mining. In fact, our approach can be extended to analyze other type of big datasets to reduce human's subjective evaluation.

7. ACKNOWLEDGEMENT

First author would like to thank the Applied Mathematics REU 2016 program at University of California, Los Angeles where his topic modeling experienced originated and where he was given access to the datasets used in this paper.

8. REFERENCES

[1] D. Kuang, J. Choo, and H. Park, "Nonnegative matrix factorization for interactive topic modeling and document clustering," in *Partitional Clustering Algorithms*, pp. 215–243, Springer, 2015.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *Advances in neural information processing systems*, pp. 601–608, 2002.

[3] R. Tripodi, S. Vascon, and M. Pelillo, "Context aware nonnegative matrix factorization clustering," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 1719–1724, IEEE, 2016.

[4] W. Wei and J. A. Gulla, "Sentiment learning on product reviews via sentiment ontology tree," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 404–413, Association for Computational Linguistics, 2010.

[5] Y. Lu and C. Zhai, "Opinion integration through semi-supervised topic modeling," in *Proceedings of the 17th international conference on World Wide Web*, pp. 121–130, ACM, 2008.

[6] W. Li and A. McCallum, "Pachinko allocation: Dag-structured mixture models of topic correlations," in *Proceedings of the 23rd international conference on Machine learning*, pp. 577–584, ACM, 2006.

[7] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, pp. 288–296, 2009.

[8] D. Greene, D. OCallaghan, and P. Cunningham, "How many topics? stability analysis for topic models," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 498–513, Springer, 2014.

[9] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring topic coherence over many models and many topics," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 952–961, Association for Computational Linguistics, 2012.

[10] S. S. Kataria, K. S. Kumar, R. R. Rastogi, P. Sen, and S. H. Sengamedu, "Entity disambiguation with hierarchical topic models," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1037–1045, ACM, 2011.

[11] V. Perrone, P. A. Jenkins, D. Spano, and Y. W. Teh, "Poisson random fields for dynamic feature models," *Journal of Machine Learning Research*, vol. 18, no. 127, pp. 1–45, 2017.

[12] C. Robert, "Machine learning, a probabilistic perspective," 2014.

[13] L. M. S. Dukler, Han and Wang, "Social media data analysis," 2016.

[14] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 1031–1040, ACM, 2011.