Topic Discovery and Evolution Through Social Media

Zachary D. Vink
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
vinkx009@morris.umn.edu

ABSTRACT

Topic discovery and topic evolution are fields of study with continuous development over the past two decades. Topic discovery is the process of labeling a document with a set of topics which accurately describe the purpose of the document. Topic evolution is the description of changes within a set of features showing how those features describe topics differently or similarly over a period of time. Topic evolution began its development at the turn of the century, and modifications to the processes behind topic evolution increased rapidly with aid from social media. Today, topic evolution utilizes the optimal methods behind topic discovery to lay the foundation for its algorithms. I discuss the major roots for topic discovery and its latest modifications via social media in the current paper. Then show the improvements social media has granted to enable the topic evolution algorithm LTECS (Learning Topic Evolution from Content and Social media activity). LTECS is an approach for both topic discovery and topic evolution.

Keywords

Topic Evolution, Topic Discovery, Badge Model, Social Media, Dirichlet, LTECS

1. INTRODUCTION

For our purposes a document is a *collection of words* that have one or more key ideas. A collection of documents is known as a **corpus**. The ideas in each document are known as topics. Topic discovery is the process of revealing topics describing a document. Topic evolution is an evaluation of changes in topic definitions over time. We use social media outlets for optimizing topic discovery and evolution algorithms. Social media users share a variety of documents, and connect the world through sharing, blogging and posting about their interests. This set of activities allows for large scale data collection. The documents *and their context* is used in topic discovery and evolution. Access to information around these documents lead to more accurate results.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/us/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
UMM CSci Senior Seminar Conference, December 2015 Morris, MN.

For example, if an article is shared only by users who label themselves as chefs or cooks, we can determine there is a high probability that the article is about food (see section 4 for more details). This example is topic discovery through social media. Topic discovery has evolved immensely since the invention of Latent Dirichlet Allocation (LDA). LDA will be discussed further in section 3. Today, most methods of topic discovery use variations of LDA or Non-Negative Matrix Factorization (NMF). NMF will be covered in depth in section 2. This paper reviews The *badge model* for an example of topic discovery using NMF. The badge model will be covered in section 4.

Social media as a data outlet has consistent amounts of documents shared month to month [1]. This is an advantage for testing topic evolution models. Topic evolution is the description of changes within a subset of features showing how those features describe topics differently or similarly over a period of time. For example, an article printed in the 1920's describes an increase in power for the national government. Using a method of topic discovery we determine this article is about Republicans. However, if we modify the topic discovery algorithm to compare it to articles from the 21^{st} century, it would label the document as Democratic. The feature "big government" has changed which party it is associated with over time. The description of this change is its topic evolution. In the current paper I review the topic evolution method LTECS (Learning Topic Evolution from Content and Social media activity) in section 5.

2. BACKGROUND

A **vocabulary** is each word in the corpus we are using. Stemming is the process of reducing words to their base form. For example, the words *exercising*, *exercised*, and *exercises* can all be stemmed to the word *exercise*. For our purposes when we refer to a word in a vocabulary, we are referring to all words that can be stemmed to the given word.

A model is a system used to describe a set of observations. We use a model as a tool for describing a corpus. A model is created by using a sample of documents from the corpus. Once a proper model is created, we can draw conclusions about other documents in the corpus that were not part of the sample. The methods covered in this paper use an iterative approach for model creation. An iterative approach works by creating multiple models and using the one with the least amount of error. The process for determining the amount of error is dependent on the type of model. Probabilistic models evaluate the probability of the model being a possible description of the corpus. A higher probability means less error in the model.

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Figure 1: Each column represents a topic. The words in the column are the words LDA used to generate that topic. The topic names were chosen by hand by looking at the columns.

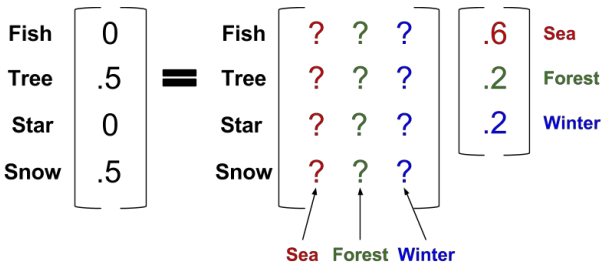


Figure 2: The matrix in the center shows associations between words and topics. This is a dictionary.

A loss function is how matrix models determine the amount of error in the model. A dictionary is a matrix creating an association between words and topics. First, we describe the process of dictionary creation through NMF, then show how a loss function determines the amount of error. In figure 2 the matrix on the left describes a document. The center matrix is the dictionary we are attempting to create, and the matrix on the right is the weights of each topic describing the document. The given document can be described as:

- Half of the words in the document are stemmed to *Tree* and the other half to *Snow*.
- The document is described most by the word *Sea* (60%), and is also described by the words *Forest* (20%) and *Winter* (20%).

Solving for the center matrix gives us a dictionary associating the words *Fish*, *Tree*, *Star*, and *Snow*, to the topics *Sea*, *Forest*, and *Winter*. However, this dictionary would not be very accurate, because it is based on one document. The process of using many documents to create a dictionary is NMF. NMF begins by taking a sample of documents from a corpus and assigning them topics (through a process like LDA, see section 3). Then we solve for a dictionary. However, it is highly unlikely to find a dictionary that works. Instead we solve for a dictionary with the least error.

An l_2-norm is the square root of all the entries squared and added together. This value enumerates the amount of error for the dictionary. NMF discovers the dictionary with the lowest l_2-norm. However, this process is very computationally heavy. Different forms of NMF alter the process for discovering the minimum error as seen in sections 4 and 5. The result of altering the l_2-norm of a function is the loss function for the model.

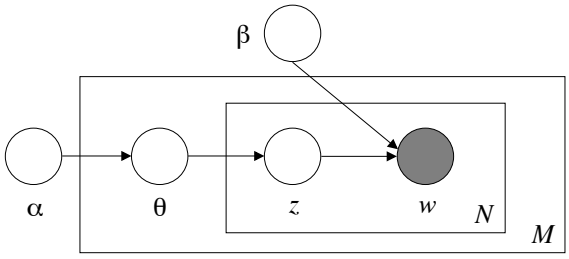


Figure 3: This template shows the connection between parameters and variables in the LDA Model.

3. LATENT DIRICHLET ALLOCATION

LDA works as an iterative process. The goal is to find out which topics have the highest probability of being the topics that helped create the paper. Unlike the examples mentioned previously, LDA describes a topic as words that are correlated with each other. Figure 1 shows potential results from using LDA to get a document's topics. LDA works under the assumption that three dependencies held true when the document was created.

1. A document was created with a topic distribution describing the topic proportions within the document.
2. Each topic has a word distribution describing the word proportions within the topic..
3. Each word in the document was created based on the topic distribution, and the word distribution.

The process begins by stemming, then removing words that will not be useful for discovering topics. For example, words such as *The*, *For*, *A*, *By*, and *As* are used frequently across all topics. We would remove these words from the vocabulary. However, words such as *Fish*, *Tree*, *Car*, and *Oxygen* are important to identifying topics. These words would remain in the vocabulary.

We set $\beta_{1:K}$ equal to the word distribution for each topic. These distributions are random to begin so we use Dir(V) to create the word proportions with a Dirichlet distribution (V is a vector the size of the vocabulary). Let K denote the number of topics in the corpus. We set θ to an $M \times K$ matrix, where M is the number of documents in the corpus. Then for each row in θ we pass α (a K sized vector) to the function Dir(α) to create a Dirichlet distribution for each row z in θ.

We use figure 3 to understand the dependencies and parameters we are attempting to discover. Recall the three assumptions made earlier for LDA. The first corresponds to the parameter z. In figure 3 we see this as a blank circle, this means it is a parameter we are attempting to discover. The same is true for β (assumption 2). We are using the third assumption to create our model. If we can discover a β and θ that result in generating our corpus, then we have discovered model used to create the corpus. However, given

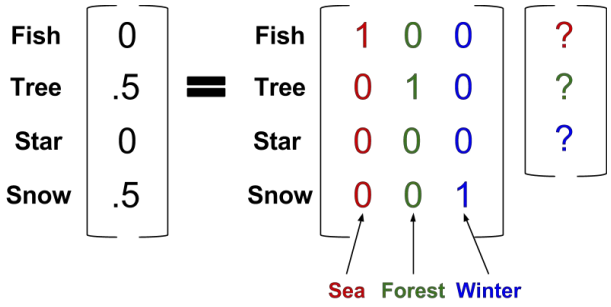


Figure 4: The figure shows an example of $\mathbf{y}_i \approx \mathbf{B}\theta_i$. Where our dictionary B is known and we are attempting to find θ for document i

the assumptions that are made, we know this is impossible [1]. Instead we attempt to find a model that has the highest probable chance of creating the corpus. This is done by editing the parameters for the Dirichlet distributions numerous times. After each time, we check the probability of the corpus being generated under those parameters. This is repeated until skewing the parameters continuously results in a lower probability. Then, the model with the highest probability is used.

LDA has a major strength and weakness as a topic discovery model. Through it's simplicity LDA allows for building more specific probabilistic models for handling specific corpora. The weakness of LDA is its assumption on context accuracy. The use of the word mouse might be considered a sign of the topic computer. However, it could also be referring to the rodent in the paper. If enough anomalies like the previous example exist in the corpus, then the results of LDA will be skewed [1].

4. THE BADGE MODEL

The badge model uses social media to improve accuracy for topic discovery. The badge model utilizes descriptions of users to predict topic labels for documents shared by those users. Unlike LDA, the badge model uses matrix factorization to determine topic labels for a document. The badge model uses matrix factorization to train a dictionary. Then this dictionary is used to determine the topics for a document.

The badge model operates under the assumption that there is a set of users who are associated with a document. This association, we assume, means a word describing the user also has a weight on the potential topic of the document [5]. We call a word describing a user a *tag*. For example, if a document is read by 100 people, 30 of which have the tag chef, and 85 have the tag vegetarian, we can conclude that the article is about vegetarian food. In this example, we also see that the proportion of each tag leaves an influence on the interpretation. While 85 readers were vegetarians, only 30 of the total readers were chefs. This suggests the article is more focused on diet restriction. If the numbers were reversed we would assume it is a complicated recipe for chefs, that happens to be vegetarian. The badge model returns topics with weighted results so we can draw these conclusions [2].

The badge model uses matrix factorization to determine the weight of each topic. It first takes each word in every document in the training corpus as the training vocabulary. It then starts training a dictionary represented as a matrix **B**. This is a V x K matrix with V rows representing each word in the vocabulary, and K potential topics that are associated with these words. This dictionary is used in equation 1 as the base for the badge model.

$$\mathbf{y}_i \approx \mathbf{B}\theta_i \tag{1}$$

The badge model takes document i represented by weights in $\mathbf{y}_i$. In Figure 4 the document i is described by the word tree and snow equally, and is not described by fish. The vector $\mathbf{y}$ is a V sized vector containing an entry representing each word in the corpus vocabulary. It sets this equal to the dictionary matrix **B** multiplied by the vector θ_i. The vector θ_i is a weight of the "badges", or topics, used to describe document i. Figure 4 shows a completed dictionary associating each word with a topic. The goal is to find a θ_i to complete the formula.

For the badge model to succeed in labeling a document, it has to train a dictionary beforehand. This process must solve a key problem with user to document association. The matrix **B** is a representation of weights for topics associated with a variety of documents. However, to create this dictionary we assume the set of badges describing a user also describes a document shared by the user. While this assumption is reasonable, it is not intended to include all the badges describing the user. Let's look at the vegetarian example from before to understand the issue. While almost every reader was labeled as a vegetarian, these users were not only labeled as vegetarians. Some were labeled as Lawyers, Chefs, Mothers, Fathers, etc. The important badges will show up in more individuals than the badges not describing the document. It is this description that must be upheld when training the dictionary.

To train the dictionary, the badge model uses a loss objective to determine the best dictionary. A loss objective approach means it will create numerous variations of the dictionary and choose one that is the most accurate. Accuracy is determined by minimizing the amount of error as described in section 2. To determine the most accurate dictionary we use the loss objective function:

$$\min_{B\geq 0} \sum_{i=1}^{N} l(\mathbf{y}_i, \mathbf{B}\theta_i) + \lambda_B \sum_{j=1}^{V}\sum_{k=1}^{K} |\mathbf{B}_{jk}| \tag{2}$$

The function above takes several parameters: the potential dictionary matrix **B**, the corpus of documents, the badges used to describe users sharing the document, and a sparsity promoter parameter. The corpus of documents is broken down into weighted vectors, where $\mathbf{y}_i$ is the weights for document i. Each document is iterated over in the left side of the summation to use as a parameter in the l function. The l function the l_2-norm described in section 2. This function also uses the vector of user badges associated with that document, and the dictionary matrix as well. The last parameter λ, is the sparsity promoter. The value of λ is increased if we would like to penalize matrices that are not sparse. For an in-depth solution to equation 2 use the supplemental material for [2]. Solving equation 2 by optimizing over the matrix **B** gives us the final dictionary with the least error.

The dictionary training is where the badge model stands out from other forms of NMF methods for topic discovery. Older forms of NMF use methods such as LDA to obtain a

(a) Olympics (#2) (b) soccer (#5) (c) Labour (#10) (d) views (#27)

Figure 5: Each set of words shows how strongly they describe the correlated badge with its size.

θ for each document. Recall θ is the weight of each topic (or badge) describing its associated document. The badge model differs by taking all of the badges describing users who shared the article and weighting them to create θ. For example, take the users with their badges below:

- User 1: Liberal, Minnesotan
- User 2: Liberal, Athlete
- User 3: Conservative, Athlete

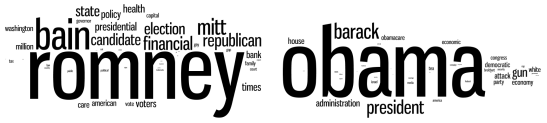
The θ for the document shared by these three users would have a weight of 1/3 for the topic *Liberal*, 1/3 for Athlete, 1/6 for Conservative, and 1/6 for Minnesotan. The benefits of using topics based off of user descriptions are discussed further later in this section.

After a dictionary has been created, the badge model uses a similar function to equation 2 optimized over the badge vector θ_i. The following equation is used when analyzing a document for topics.

$$\min_{\theta \geq 0} ||\mathbf{y}_i - \mathbf{B}\theta_i||_2^2 + \lambda_\theta ||\theta_i||_1 \tag{3}$$

Equation 3 promotes efficiency with values of λ close to one or zero when solving. This promotes sparsity in the badges vector. As a result when a document is labeled using the badge model it has varying weights to accurately describe stronger topics in a document, and we avoid getting results with numerous topics with low weights.

Now we look at examples using the badge model to review its strengths and weaknesses. El-Arini *et al* provide a well described experiment they performed using the badge bodel and Twitter as its source of data. Twitter is a data heavy social media outlet that allows for ease of use with the badge model. It is easy to use with the badge model because users have tags that describe them. These tags will be used as the badges for training the dictionary. Twitter also has an open API for obtaining a random sample of tweet. El-Arini *et al* use the Twitter Garden Hose which supplies a random sample of tweet in a specified time. Using the API, El-Arini *et al* get over 120 million tweets from over 40 million users. The tweets come from September 2010, September 2011, and September 2012. For focusing the data, they then cut all tweets that are not shared news articles. They limit the articles to a set of 20,000 potential news outlets. This left over one million shared articles for each of the time sets. They then trained a dictionary for each period and used the dictionary for labeling a set of articles from *The Guardian*. In total they tested 14,000 articles. When they created the dictionaries El-Arini *et al* used 4,460 unique badges in September 2010, 5,029 badges in September 2011, and 5,247 badges in September 2012.



(a) progressive > tcot (b) tcot > progressive

Figure 6: The figure shows liberal references describing a Republican badge and vice versa.

El-Arini *et al* first analyze the resulting dictionaries. This is important to see if the words associated with certain badges are accurate. It is likewise important to discover weights for words symbolizing how much association they have with the given badge. In Figure 1 we can see the relative weights of words describing some of the most used badges from the training set. We can see through Figure 5 (a) that the badge *Olympics* is strongly associated with words such as *olympic*, *paralympics*, *athletes*, and *london*. This shows that the dictionary was correctly weighting how to describe the topic *Olympics*. The examples seen in Figure 5 (b) and (c) also show accurate descriptions of words used to describe the badge. However, (d) shows a mixed conglomeration of words describing the badge *view*. El-Arini *et al* note that this occurs only twice in the top one hundred badges. This describes the accuracy of using the badge model. However, it shows there is room for improvement in the badge selection process used in the experiment [2].

Another area for improvement in the badge model can be seen from using the dictionary based off articles in September 2012. If we look at the tags *Liberal* and *tcot* (**T**op **C**onservatives **O**n **T**witter) we see unusual results. In Figure 6 we see the opposite of what is expected. This is explained using the context of when the dictionary is created. The election of 2012 sparked a dramatic increase in slandering the opposing party in articles. This resulted in articles typically shared by republicans actually being about liberal topics [2].

5. LTECS

Topic evolution expands on document labeling. The primary method this paper covers is called Learning Topic Evolution from Content and Social media activity (LTECS). LTECS uses Non-negative Matrix Factorization for labeling documents. However, compared to the badge model (covered in section 4) LTECS implements collective factorization in order to make predictions from two matrices at the same time. LTECS implements collective factorization to compare and contrast describing documents through a trained

corpus of labels and by using a community of users that are connected to the document (via *sharing* or *tweeting*). In conjunction with data over time, LTECS uses this information to determine if topics change associations with either topics or users.

Symbol	Description
t	an arbitrary time
d	a document in the corpus for training
f	a textual feature in a document, typically a non-stop word
k	the number of topics describing all the documents in the training corpus
N_d^t	the number of documents in the corpus associated with time t
N_f	the number of textual features in the corpus associated with time t
N_u	the number of users who shared a document at time t
$\mathbf{W}^{\mathrm{t}}$	An N_d^t x k matrix
$\mathbf{H}^{\mathrm{t}}$	An k x N_f matrix
$\mathbf{G}^{\mathrm{t}}$	An k x N_u matrix
$\mathbf{X}^{\mathrm{t}}$	An N_d^t x N_f matrix
$\mathbf{U}^{\mathrm{t}}$	An N_d^t x N_u matrix

Figure 7: The table can used for symbol references in understanding the LTECS method

To begin understanding LTECS, we look at the necessary matrices. $\mathbf{X}^{\mathrm{t}}$ and $\mathbf{U}^{\mathrm{t}}$ denote two matrices defined at time t [3]. The matrix $\mathbf{X}^{\mathrm{t}}$ is an N_d^t x N_f matrix at time t composed of N_d^t documents and N_f textual features. Each row in $\mathbf{X}^{\mathrm{t}}$ represents a single document d that was shared (through an arbitrary social media site) at time t. The document is described by one or more of the textual features f. The textual features that compose the columns in $\mathbf{X}^{\mathrm{t}}$ represent a variety of attributes semi-unique to the given document. Here, semi-unique is used to describe an attribute that is rare enough among all the documents that it has a possible impact on the actual identity of the article. The matrix, when created, is a variety of weights that show which labels are associated to a given document. The matrix $\mathbf{U}^{\mathrm{t}}$ is similar to $\mathbf{X}^{\mathrm{t}}$ except the textual features are replaced by the users who shared a document at time t. $\mathbf{U}^{\mathrm{t}}$ is used to help create a connection between topics and users so we can label a document in terms of the people who shared it. The result is a N_d^t x N_u matrix. The documents used to describe $\mathbf{U}^{\mathrm{t}}$ and $\mathbf{X}^{\mathrm{t}}$ are the same documents in the same arbitrary order. However, the amount of textual features for $\mathbf{X}^{\mathrm{t}}$ does not need to match the number of users for $\mathbf{U}^{\mathrm{t}}$.

As mentioned previously, $\mathbf{U}^{\mathrm{t}}$ and $\mathbf{X}^{\mathrm{t}}$ are both used by LTECS to label topics. To use these matrices for topic discovery LTECS uses the standard Non-negative Matrix Factorization (NMF) technique. LTECS uses a trained matrix $\mathbf{H}^{\mathrm{t}}$ to define $\mathbf{W}^{\mathrm{t}}$.

$$\mathbf{X}^t \approx \mathbf{W}^t\mathbf{H}^t \tag{4}$$

In equation 4, $\mathbf{H}^{\mathrm{t}}$ is a k x N_f trained matrix used to create $\mathbf{W}^{\mathrm{t}}$. The number of topics describing the features and documents may change depending on the set of documents. To accommodate, LTECS makes k a parameter to increase usability. $\mathbf{W}^{\mathrm{t}}$ is a N_d^t x k matrix. This matrix associates

$$\underset{\mathbf{W}^t}{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_d,1} & a_{N_d,2} & \cdots & a_{N_d,k} \end{bmatrix}} \underset{\mathbf{H}^t}{\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,N_f} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,N_f} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k,1} & b_{k,2} & \cdots & b_{k,N_f} \end{bmatrix}}$$

Figure 8: The equation above shows how $\mathbf{X}^t$ is formed from the matrices $\mathbf{W}^t$ and $\mathbf{H}^t$

a document with a variety of topics based on the highest weights in the matrix. The matrix representations below show how these matrices are filled.

Recall that $\mathbf{W}^{\mathrm{t}}$ has rows equal to the number of documents and columns equaling the topics we have to choose from as seen in figure 5. Here, if the values for $a_{2,2}$ and $a_{2,3}$ are equal to one, then we know that the topics associated with column 2 and 3 perfectly describe document 2. To obtain the values that fill $\mathbf{W}^{\mathrm{t}}$ we decompose $\mathbf{H}^{\mathrm{t}}$ with relation to $\mathbf{X}^{\mathrm{t}}$. This process is the same NMF we used to describe the badge model in section 4. The matrix $\mathbf{U}^{\mathrm{t}}$ is used to decompose a matrix similar to $\mathbf{H}^{\mathrm{t}}$ in order to relate documents to users instead of textual features.

$$\mathbf{U}^t \approx \mathbf{W}^t\mathbf{G}^t \tag{5}$$

LTECS uses collective factorization to find a $\mathbf{W}^{\mathrm{t}}$ that fulfills both the trained data from $\mathbf{G}^{\mathrm{t}}$ and $\mathbf{H}^{\mathrm{t}}$. For a better understanding of how collective factorization works see [4].

The purpose of LTECS is also to model how topics evolve over time [3]. In order to satisfy this condition equation 4 is modified to include the topic evolution matrix $\mathbf{M}_T^t$. This matrix is used to describe how the topics change over time. If the topic evolution matrix is close to the identity matrix then the topics represent nearly the same textual features from time $t-1$ to time t. Adding the topic evolution matrix to equation 4 yields the following:

$$\mathbf{X}^t \approx \mathbf{W}^t\mathbf{M}_T^t\mathbf{H}^{t-1} \tag{6}$$

$$\mathbf{U}^t \approx \mathbf{W}^t\mathbf{M}_T^t\mathbf{G}^{t-1} \tag{7}$$

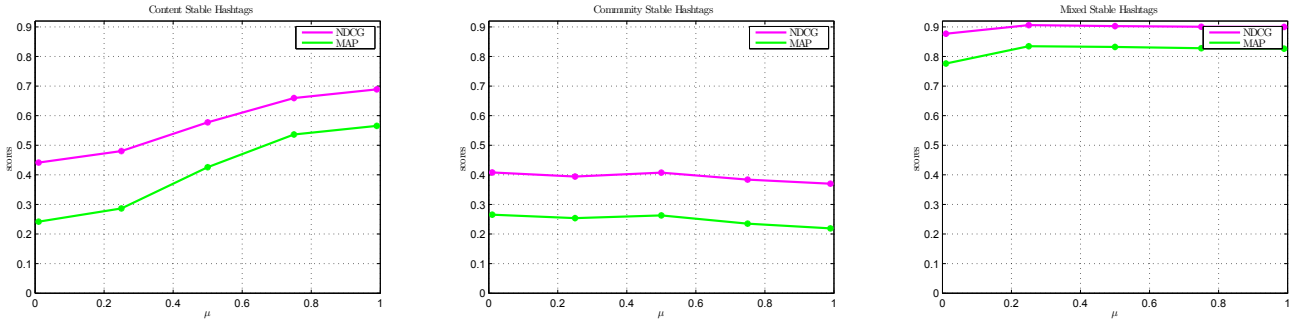
This is also done to equation 5 to accurately represent the topic evolution for the community of users resulting in equation 7. When using the equation we assume $\mathbf{H}^{t-1}$ is known when computing information for $\mathbf{H}^t$. The product of the topic evolution matrix with $\mathbf{H}^{t-1}$ will produce $\mathbf{H}^t$. This linear combination is the key to discovering the topic evolution matrix. If $\mathbf{H}^t$ and $\mathbf{H}^{t-1}$ are known then we can solve the following equation for $\mathbf{M}_T^t$.

$$\mathbf{H}^t \approx \mathbf{M}_T^t\mathbf{H}^{t-1} \tag{8}$$

LTECS relies heavily on assumptions for correlations between users, content, and topic discovery. These assumptions mean there is a likely chance for error in many cases. In order to determine the best topic evolution matrix and $\mathbf{W}^t$ from equations 6 and 7 LTECS uses a loss function to optimize the results. The LTECS loss function is:

$$L = \mu L_T + (1-\mu)L_C + R \tag{9}$$

The loss function above is used similarly to the objective function in section 4. The L_T represents the accuracy of



(a) content stable; $k = 5$ (b) community stable; $k = 5$ (c) mixed stable; $k = 5$

Figure 9: The graphs above show the accuracy of LTECS based on two types of measurements (NDCG and MAP) [3]

content based labeling of the document (solving equation 4), while L_C is the accuracy of labeling based on the community sharing the document (solving equation 5). The parameter μ is set to put emphasis on creating either a a content accurate or community accurate result. The complexity of optimizing the loss objective is covered in depth in [3].

LTECS has two main purposes: topic discovery, and determining if topics in a corpus are *content stable*, *community stable*, or *mixed-stable*. *Content stable* means that each textual feature continues to describe the same topic over time, while the relationship between each user and its topic changes. *Community stable* is the opposite of *content stable*, and *mixed-stable* is when both content and community are stable. A content stable topic is expected to be more common the community stable [3]. To determine if LTECS can accurately describe community vs. content stability we review a study conducted by Kalyanam *et al.*

The study began by collecting data from 80 different news sources via twitter. Kalyanam *et al.* then filtered down the information based off of missing information or document type to obtain 33,387 articles. These articles were described by 384 hashtags. *Ground truth* is the information directly observed rather than inferred [6]. LTECS uses the hashtags associated with each article as the *ground truth* for the study. This means that the results of using LTECS should be reasonably close to the hashtags describing each article to be considered accurate. Figure 9 shows the scores of LTECS in the experiment completed by Kalyanam *et al.* The score is a weighted accuracy of the returned topics when compared to the ground truth. The results show higher accuracy following μ's value as expected. With content stable topics we see higher accuracy at high values of μ. This is not seen as strongly in community stable topics. Kalyanam *et al.* propose this is based on the data used [3]. Figure 9 shows the weakness in LTECS. Given the focus on topic evolution the accuracy of the general topic discover was weakened. Kalyanam *et al.* conclude by recommending further work on using more accurate forms of topic discovery to determine topics before evaluating the relationship between topics and users or features.

6. CONCLUSION

Reviewing three separate methods for Topic Discovery and Topic Evolution reveals the complexity of optimization. By looking at LDA we discovered the strengths and weaknesses for probability based topic discovery. Given LDA's weaknesses, the badge model and LTECS utilized social media for creating inferences about how we view a document. The badge model focused its strengths on sparsity for efficiency while yeilding accurate results. This is in contrast to LTECS which made strides to recognizing a topics evolution while determining if it is more accurate to describe a document via users (like the badge model) or through document content (like LDA). LTECS uses collective factorization to accomplish these discoveries which differs significantly from the methods used by LDA and the badge model. There is likely to be improvements on all of these models in the future while attempting to create a perfect method for Topic Discovery and Evolution.

7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[2] K. El-Arini, M. Xu, E. B. Fox, and C. Guestrin. Representing documents through their readers. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 14–22, New York, NY, USA, 2013. ACM.

[3] J. Kalyanam, A. Mantrach, D. Saez-Trumper, H. Vahabi, and G. Lanckriet. Leveraging social context for modeling topic evolution. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 517–526, New York, NY, USA, 2015. ACM.

[4] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.

[5] O. Tsur and A. Rappoport. What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 643–652, New York, NY, USA, 2012. ACM.

[6] Wikipedia. Ground Truth — Wikipedia, The Free Encyclopedia, 2015. [Online; accessed 10-November-2015].