THEME SUPERVISED NONNEGATIVE MATRIX FACTORIZATION FOR TOPIC MODELING

by

Burak Suyunu

B.S., Computer Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2020

ACKNOWLEDGEMENTS

After I started writing this thesis, many different and interesting events took place both in the outside world and in my world. I hope that after this thesis is defended, our world will begin to transform into a healthier and more livable place.

First of all, I would like to thank my family, especially my mother and father, for being patient and supportive in every sense. They witnessed firsthand my struggle and always tried their best to create a distraction-free environment for me. I also send the biggest hugs to my niece and nephews, who didn't even knock on my door because their uncle was writing his thesis.

I would also like to thank my fiancee Özlem, whom I met during my dissertation period and hopefully will marry soon, for enduring me and trying a thousand different ways to support me in this process. I feel like she is the one :)

I am grateful to all my professors in our department, especially Ali Taylan Cemgil, Tunga Güngör, and Suzan Üsküdarlı, for their guidance and help throughout my master's degree.

And finally, I would like to thank my friends, whom I consider myself very lucky to have after my family (and my fiancee). Our online chitchats, games, and celebrations were some of the most fun times I had in this ridiculous year.

ABSTRACT

THEME SUPERVISED NONNEGATIVE MATRIX FACTORIZATION FOR TOPIC MODELING

Topic models are often used to organize and interpret large and unstructured corpora of text documents. They try to explain the topics that constitute the semantic infrastructure of the document sets and try to find the distributions of these topics for the documents. Because of its unsupervised nature, the outputs of a topic model has to be interpretable to represent its success. However, the results of a topic model are usually weakly correlated with human interpretation. In this thesis, we propose a semi-supervised topic model called *Theme Supervised Nonnegative Matrix Factorization* that can benefit from labeled documents to improve and facilitate the interpretation of the topics. Our model constrains the representation of the topics to align with the labeled documents and this enables the topics discovered by the model to be readily understood. To utilize the labels provided by the documents more efficiently and to explore the document sets in more depth, we used a hierarchical topic structure consisting of themes, subtopics, and background topics in our model. We created layers under the themes that permit unsupervised learning for subtopics. This hierarchical structure, with the unsupervised learning capability it provides, enables our model, which was restricted with supervision, to discover new dimensions and make more detailed classifications. We tested our model on Schwartz dataset we created, as well as Brown and Reuters datasets with different supervision ratios.Our model estimates the topics of the documents much better than the traditional nonnegative matrix factorization and latent Dirichlet allocation for any situation; and besides, the effect of supervision is noteworthy, especially at low ratios. Moreover, our new term scoring metric successfully alters the weights of significant and insignificant terms for each topic and makes the topics easier to understand and interpret.

ÖZET

KONU MODELLEME İÇİN TEMA DENETİMLİ NEGATİF OLMAYAN MATRİS AYRIŞTIRMASI

Konu modelleri, büyük ve yapısal olmayan yazılı döküman setlerinin organize edilip yorumlanmasında sıklıkla kullanılır. Doküman setlerinin anlamsal altyapısını oluşturan konuları açıklamaya ve bu konuların dokümanlar üzerindeki dağılımlarını bulmaya çalışırlar. Denetimsiz doğası nedeniyle, bir konu modeli başarısını gösterebilmesi için, çıktılarının yorumlanabilir olması gerekir. Fakat, bir konu modelinin sonuçları genellikle insan yorumuyla zayıf bir şekilde ilişkilendirilir. Bu tezde, konuların yorumlanmasını iyileştirmek ve kolaylaştırmak için etiketli belgelerden yararlanabilen, *Tema Denetimli Negatif Olmayan Matris Ayrıştırması* adlı yarı denetimli bir konu modeli öneriyoruz. Modelimiz, konuların temsilini etiketli belgelerle eşleşecek şekilde kısıtlar ve bu, model tarafından keşfedilen konuların kolayca anlaşılmasını sağlar. Dokümanların sağladığı etiketleri daha verimli kullanabilmek ve doküman setlerini daha derinlemesine inceleyebilmek için, modelimizde temalar, alt konular ve arka plan konularından oluşan hiyerarşik bir konu yapısı kullandık. Temaların altında, alt konular içi denetimsiz öğrenmeye izin veren katmanlar olşturuyorduk. Bu hiyerarşik yapı, kendi içinde sağladığı denetimsiz öğrenme kabiliyeti ile, denetim ile kısıtladığımız modelimizin yeni boyutlar keşfedip, daha detaylı sınıflandırmalar yapabilmesine olanak sağlar. Modelimizi, oluşturduğumuz Schwartz veri kümesinin yanı sıra Brown ve Reuters veri kümelerinde farklı denetim oranlarıyla test ettik. Modelimiz, belgelerin konularını geleneksel negatif olmayan matris ayrıştırmasından ve gizli Dirichlet tahsisi'nden her koşulda çok daha iyi tahmin ediyor; ve bunun yanında, denetimin etkisi bir logaritmik fonksiyon gibi davranır ve daha düşük oranlarda en fazla etkiye sahiptir. Ayrıca yeni terim puanlama metriğimiz, her konu için önemli ve önemsiz terimlerin ağırlıklarını başarıyla değiştirerek konuların anlaşılmasını ve yorumlanmasını kolaylaştırır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
1.1. Motivation . . . 2
1.2. Contributions . . . 5
1.3. Thesis Outline . . . 6
2. BACKGROUND AND RELATED WORKS . . . 7
2.1. Topic Modeling . . . 7
2.2. Nonnegative Matrix Factorization . . . 9
2.2.1. Cost Function: Square of the Euclidean Distance . . . 12
2.2.2. Cost Function: Kullback-Leibler divergence . . . 12
2.3. Latent Semantic Analysis . . . 13
2.4. Probabilistic Latent Semantic Analysis . . . 14
2.5. Latent Dirichlet Allocation . . . 16
2.6. Relations Between NMF, PLSA, and LDA . . . 18
2.7. Related Extensions . . . 19
3. THEME SUPERVISED NONNEGATIVE MATRIX FACTORIZATION . . . 22
3.1. Step 1: Traditional Nonnegative Matrix Factorization . . . 22
3.1.1. Demonstration . . . 24
3.2. Step 2: Supervision . . . 25
3.2.1. Demonstration . . . 26
3.3. Step 3: Themes and Subtopics . . . 28
3.3.1. Demonstration . . . 29

3.4. Step 4: Background Topic 30
3.4.1. Demonstration 31
3.5. Step 5: Separation 31
3.5.1. Demonstration 32
3.6. Extension: Fully Supervised 33
3.6.1. Demonstration 34
3.7. Extension: New Topics 35
3.7.1. Demonstration 35
4. BCOOL INITIALIZATION 38
5. SCORING OF DOCUMENTS 41
5.1. Theme Score 43
6. SCORING OF TERMS 46
6.1. Purity and Theme Term Score 46
6.1.1. Scoring Terms for Each Document 49
7. PREPARATION FOR EXPERIMENTS 52
7.1. Text Preprocessing 52
7.2. Datasets 52
7.2.1. Brown Corpus 53
7.2.2. Reuters Corpus 53
7.2.3. Schwartz's Theory of Basic Human Values Dataset 53
7.3. Evaluation Metrics 55
7.3.1. Log Rank Accuracy 56
7.3.2. Genetic Algorithm for Traditional NMF and LDA 58
8. EXPERIMENTS AND RESULTS 62
8.1. Finding The Best Parameter Settings 63
8.2. How to Initialize: bCool vs Random 63
8.3. How to Model: Novel vs Traditional 67
8.4. How to Interpret: Themes vs Topics 71
9. CONCLUSION AND DISCUSSION 78
REFERENCES 82

LIST OF FIGURES

Figure 2.1. Illustration of NMF in terms of linear approximation of X 10

Figure 2.2. Plate notation for the PLSA model (asymmetric formulation). The document D and the term T are observable variables, the topic Z is a latent variable. 15

Figure 2.3. Illustrations of asymmetric and symmetric formulation processes of PLSA. 16

Figure 2.4. Plate notation for the LDA model. 17

Figure 3.1. Demonstration of Step 1: Traditional Nonnegative Matrix Factorization 24

Figure 3.2. Demonstration of Step 2: Supervision 27

Figure 3.3. Demonstration of Step 3: Themes and Subtopics 30

Figure 3.4. Demonstration of Step 4: Background Topic 31

Figure 3.5. Demonstration of Step 5: Separation 33

Figure 3.6. Demonstration of the fully supervised version of *TSNMF* 36

Figure 3.7. Demonstration of exploring new topics with *TSNMF* 37

Figure 7.1. The relationships among the basic human values and higher order groups in Schwartz's Theory of Basic Human Values. 54

Figure 7.2. Change in the *log rank accuracy* with respect to the change in the ranking of an assigned theme in a dataset with 20 themes. 58

Figure 7.3. Two-point crossover 60

Figure 7.4. Shift change mutation 61

Figure 8.1. Difference between *log rank accuracy* of *bCool* and random initialized models on all datesets. 65

Figure 8.2. Plots of *log rank accuracy* of *TSNMF* models and traditional NMF-LDA on all datasets with respect to the change in supervision ratio. 70

Figure 8.3. The topmost 5 terms of four topics from Schwartz dataset that could be interpreted as universalism and hedonism themes for the traditional NMF model. 73

Figure 8.4. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* models with 1 subtopic and *purity* ratio of 0 and 1. 74

Figure 8.5. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* model with 3 subtopics and *purity* ratio of 0. 75

Figure 8.6. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* model with 3 subtopics and *purity* ratio of 0. 77

LIST OF TABLES

Table 5.1. Values of an example document-topic matrix. 42

Table 5.2. Normalized values of the example document-topic matrix. 42

Table 5.3. Application of theme scoring scheme on a *Separated TSNMF* example. 45

Table 7.1. Basic statistics of the datasets. 53

Table 7.2. Number of documents for each basic human value in the Schwartz dataset. 55

Table 8.1. The comparison between 3 different cost function and text encoding combinations. 64

Table 8.2. *Log rank accuracy* of *TSNMF* models and the traditional NMF-LDA models. 68

LIST OF SYMBOLS

H	Topic-term matrix or feature matrix
M or m	Number of terms
N or n	Number of documents
P or p	Number of topics
S	Supervision matrix
U	Document-topic matrix
V	Topic-term matrix
W	Document-topic matrix or coefficient matrix
X	Document-term matrix or data matrix
α	Dirichlet prior on the per-document topic distribution
β	Dirichlet prior on the per-topic term distribution
θ	Per-document topic distribution
λ	Purity ratio
Σ	Diagonal singular value matrix
φ	Per-topic term distribution
$\circ$	Hadamard product
$\{\cdot\}^T$	Transposition

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BHV	Basic Human Value
BTS	Background Term Score
DTS	Direct Term Score
EM	Expectation-Maximization
hLDA	Hierarchical Latent Dirichlet Allocation
hLLDA	Hierarchical Labeled Latent Dirichlet Allocation
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
NLP	Natural Language Processing
NMF	Nonnegative Matrix Factorization
NLTK	Natural Language Toolkit
PLSA	Probabilistic Latent Semantic Analysis
PTS	Purity Term Score
SVD	Singular Value Decomposition
tf	Term Frequency
tf-idf	Term Frequency–Inverse Document Frequency
TSNMF	Theme Supervised Nonnegative Matrix Factorization
TTS	Theme Term Score
VB	Variational Bayes

1. INTRODUCTION

In today's world, where large and unstructured data are generated continuously and often unwittingly; the need for computational methods to organize and interpret the data has become an important priority. This requires reliable and fast automatic classification and categorization methods for easier comprehension. For example, we may want to classify comments made on a website according to the sentiments, or we may want to organize our photo library according to people appearing in individual snapshots, or we may want to categorize a large corpus of articles by subject.

Supervised machine learning (ML) algorithms are now able to perform classification with very high test accuracy. However, the biggest problem of the supervised methods is that they require explicit labels, that is, they need a significant amount of high-quality manually labeled data. Although for several benchmark public datasets, pretrained models can be obtained; for specific subjects, particularly with less popular languages, labeling becomes inevitable. One other important shortcoming of the supervised methods is that they are often not very easy to interpret and trace how a particular decision is arrived at. In other words, the classification performance of supervised ML is high, but they do it as a black box.

There are also unsupervised methods, which are the version of ML that does not need labeled data at all. Although unsupervised methods do not produce as accurate classification results as supervised methods, they enable efficient analysis in areas where labeled data are low or the subject is niche. They focus on clustering or dimensionality reduction rather than classification. The main focus points of unsupervised methods are clustering and dimensionality reduction rather than classification. Examples of popular unsupervised ML algorithms are k-means [1] for clustering data, singular value decomposition (SVD) for dimensionality reduction, and topic modeling algorithms that enable analysis and categorization of written documents. Since there is no label information to measure the success of an unsupervised model, it has to

explain the data and the outputs in some way to show the success of the model.

In this thesis, we will focus on topic modeling, which is an ML and natural language processing (NLP) technique that works on written documents and whose unsupervised in nature. Topic models try to discover the semantic structure underlying a set of documents by identifying and measuring the importance of topics in the documents. Topic models represent the topics as term distributions and a topic can correspond to any subject from the most general (e.g. food) to the most specific (e.g. one-horned dinosaurs) one according to the interpretation of its term distribution. The first examples of topic models appear as latent semantic analysis (LSA) [2] and probabilistic latent semantic analysis (PLSA) [3]. But, the most widely used topic models today are latent Dirichlet allocation (LDA) [4] and nonnegative matrix factorization (NMF) [5]. Most studies in the topic modeling field are applications or extensions of NMF or LDA. In Section 2.7, we detailed some of these extensions that are related to our work.

In this thesis, we chose to extend NMF which is a dimensionality reduction algorithm that can maintain data's nonnegativity property. In the real world, it is common to see data that consist of nonnegative values such as texts, images, and audio. Factors obtained from the traditional dimensionality reduction algorithms lack interpretability for containing negative values; since it is not intuitive to use negative values while combining parts to form a whole. The Nonnegativity constraint of NMF makes the results much easier to analyze. NMF has also been successfully used in many fields such as bioinformatics [6], image processing [7, 8] and recommender systems [9].

1.1. Motivation

Topic modeling is an unsupervised method and it requires human interpretation to make its outputs meaningful. Although there is a topic distribution for each document; in order to find out what each topic corresponds to, a human expert needs to look at the term distributions of those topics and makes inferences. For example, if the term distribution of a topic is listed as puppy, kitten, cute, and panda from important

to unimportant; we can conclude that this topic is about cute animals. However, as pointed out by [4, 6, 10, 11], evaluation of topic models often leads to results that are weakly correlated with human interpretation. Because, we may not always have such clear term distributions, or we may have multiple intertwined topics. So, what if we could increase the interpretation power of topic models while having the classification power of supervised methods? We asked exactly this question to ourselves and combined the beauties of these two worlds to come up with a novel semi-supervised topic model called *Theme Supervised Nonnegative Matrix Factorization (TSNMF)*.

Semi-supervised methods do not have a clear definition like supervised or unsupervised methods do. Any method that fills the gap between unsupervised and supervised methods in different ways can be accepted as semi-supervised. Feeding labeled data information to an unsupervised method is the most common approach for developing a semi-supervised method [12]. However, since unsupervised methods do not use label information by nature, it is possible to feed this information to the model in different ways. The two most preferred approaches for converting a traditional unsupervised NMF model to a semi-supervised one are as follows:

(i) Training the model with the supervision of document labels that indicates which topics are permitted in which documents [13, 14].
(ii) Using pairwise constraints on data objects such as; must-link (the two data points must belong to the same class) and cannot-link (the two data points cannot belong to the same class) constraints [15–17].

The primary objective of *TSNMF* is to facilitate and improve the interpretation of the topics. *TSNMF* enables the user to provide documents with labels and constrains the representation of the topics to align with these labeled documents similar to the first-mentioned approach. Thus, while examining the output of the model, the interpretation of these topics will be readily understood. It is also possible to find the topic distribution of a document for any topic set, as long as enough labeled data about the topic set is collected. For example, assume that we want to estimate Twitter

users' zodiac signs from their tweets. If we directly feed the tweets into a traditional topic model and train it, we can only get common topics like sports, politics, and food. Instead, if we first collect labeled documents about zodiac signs and map topics of a *TSNMF* model to zodiac signs through these labeled documents; then we can get zodiac sign distributions of Twitter users.

Our results show that even with a small amount of supervision, we are able to estimate the topics of the documents better than the traditional NMF. On the other hand, supervised ML algorithms can probably handle this classification task with higher accuracy. But it should not be forgotten that we aim to produce results that we can interpret besides the classification, and *TSNMF* achieves both of this by compromising only a small amount of classification success.

What makes *TSNMF* different from other semi-supervised models is that it has a novel hierarchical topic structure. Instead of topics, *TSNMF* uses themes and subtopics of themes. While themes retain the semi-supervised structure of the model by replacing topics, we create layers under the themes that permit unsupervised learning for subtopics. Our aim here is to restore the ability of the model that we restricted with supervision, to discover new dimensions via unsupervised learning. Therefore, thanks to subtopics, *TSNMF* can obtain in-depth information about the themes as well as more detailed classification results for documents. If we revisit the zodiac sign example, instead of topics, now the themes correspond to the zodiac signs that we have provided with labeled documents. The subtopics of each theme corresponds to the specialized subjects of each zodiac sign. For example, for Aries, its subtopics can be about individual sports and leadership; and for Virgo, its subtopics can be about cleaning and healthy food.

1.2. Contributions

The main contributions of this thesis are summarized as follows:

(i) A novel semi-supervised topic model based on NMF called *TSNMF* is proposed. Labeled documents are utilized in the training of *TSNMF* to constrain the representation of the topics to align with the labels. It is possible to train the model fully supervised with only labeled documents or semi-supervised with the labeled and unlabeled documents together (Chapter 3).

(ii) A hierarchical topic structure consisting of themes, subtopics of themes, and background topics are introduced. This structure allows us to gain more insight about the topics and documents (Chapter 3).

(iii) A new training scheme that trains and generates a separate model for each theme is introduced. It is likened to the one-vs-all training procedure where each theme is trained versus a background topic that tries to generalize all the themes (Chapter 3).

(iv) An initialization method called *bCool* is proposed which initializes the topic-term matrix to generate more consistent factors than random initialization. Results show that *bCool* speeds up the training process at least 2 times and improves the performance for some models over random initialization. (Chapter 4).

(v) A new scoring method called *theme score* is introduced to calculate the theme distributions of documents. It uses background topics to normalize subtopic scores and chooses the highest-scoring subtopics to represent the themes (Chapter 5).

(vi) A new measure called *purity* is proposed to improve the intelligibility of term distributions via ranking and scoring terms for a particular subtopic, theme, or document. It helps us to adjust the tradeoff between the frequency and specificity of a term to generate more interpretable subtopics and themes (Chapter 6).

(vii) A new evaluation metric called *log rank accuracy* is proposed to evaluate the models. *Log rank accuracy* uses the logarithm function to calculate a score for each document according to the ranking of the themes for the document. The important property of the method is that the penalty for misclassification decreases

as the rank goes down (Chapter 7).

(viii) A genetic algorithm to find the best possible matching between the topics discovered by the traditional NMF or LDA and the predefined topics (document labels) is introduced. The objective function of the algorithm is to maximize the *log rank accuracy* for the traditional models. We were able to compare the traditional models with *TSNMF* thanks to this approach (Chapter 7).

(ix) Classification performances (log rank accuracy) of *TSNMF* models are investigated under different supervision ratios. We observed that even a small set of labeled documents can increase the performance of a topic model exponentially (Chapter 8).

(x) The contributions of the hierarchical topic structure and the *purity* measure to the interpretation of the topics are examined. The hierarchical structure helped us to define the themes in depth. *Purity* measure was able to carry the terms that are more significant to the themes to higher rankings, whereas it sent common terms such as "one" and "may" down in rankings (Chapter 8).

(xi) A Python package for *TSNMF* is made available to the public [18].

1.3. Thesis Outline

The rest of the thesis is organized as follows: Backgrounds on topic modeling, NMF, LSA, PLSA and LDA are provided and the literature on the extensions of NMF and LDA are considered in Chapter 2. Starting from the traditional NMF, we built and explained our proposed *TSNMF* model step by step in Chapter 3. An initialization method called *bCool* for *TSNMF* is proposed in Chapter 4. The theme-based scoring method for documents is introduced in Chapter 5. A new term scoring scheme with a new measure called *purity* is proposed in Chapter 6, which allows more tailored terms for each topic and also for each document. We introduced our datasets and explained evaluation metrics along with a genetic algorithm for topic matching for traditional NMF and LDA in Chapter 7. Various types of experiments are conducted and analyzed to show every aspect of our model in Chapter 8. Finally, conclusions are drawn in Chapter 9.

2. BACKGROUND AND RELATED WORKS

In this chapter, first, we will explain topic modeling in detail with examples. Then we will describe main topic modeling methods one by one. In the end, we will go over some works that are closely related to our model.

2.1. Topic Modeling

Topic modeling is a machine learning (ML) and natural language processing (NLP) technique for discovering hidden (latent) topics that occur in a set of documents. The underlying idea is that the semantics of documents are being governed by some latent variables that we do not observe. The goal of topic modeling is to uncover these latent variables - topics - that shape the meaning of documents. Topic models scan a set of documents and find the distribution of terms under latent topics that best characterize the documents using the statistics of the terms in the documents. There are two assumptions that all topic models are based on:

- Each document consists of a mixture of topics.
- Each topic consists of a collection of terms.

Traditional topic models are unsupervised ML techniques and don't use labeled documents for training. Although this situation allows the model to produce results faster with less preprocess; human interpretation takes an important role in the analysis of the results, since we do not have any prior knowledge to verify them. It is necessary to analyze the resulting term distributions of the topics to find out what each topic corresponds to. But topic models often generate results that are difficult to interpret. To remedy this problem, semi-supervised extensions of the traditional topic models were introduced (see Section 2.7 for the extensions). Semi-supervised topic models improve the interpretability of the topics by providing labeled documents to the model. Then, constrain the representation of the topics to align with the labeled documents.

This enables topics to be readily understood without the need to investigate the term distributions.

Let's see topic modeling on an example. Suppose you have the following set of sentences:

- **Sentence 1:** My sister played the flute concerto of Mozart yesterday.
- **Sentence 2:** Mozart and Schubert were the composers of music's classical period.
- **Sentence 3:** My father bought tomatoes and potatoes from the marketplace.
- **Sentence 4:** Tomatoes are fruits because they contain seeds.
- **Sentence 5:** Playing metal music to fruits accelerates their growth.

Given these sentences and asked for 2 topics, a topic model might produce something like this:

- **Sentences 1 and 2:** 95% Topic A, 5% Topic B
- **Sentences 3 and 4:** 5% Topic A, 95% Topic B
- **Sentence 5:** 60% Topic A, 40% Topic B
- **Topic A:** 25% mozart, 20% music, 10% classical, 10% metal, ... (at which point, we could interpret topic A to be about **music**)
- **Topic B:** 30% tomato, 20% fruit, 15% potato, 10% seed, ... (at which point, we could interpret topic B to be about **food**)

As a result, we got topic mixtures for sentences that contain terms with certain probabilities.

Topic modeling approaches can be divided into two categories: matrix decomposition methods which try to find a low dimensional representation of data through factorization into low-rank matrices, and probabilistic topic modeling methods, which seeks generative statistical models. An early topic model called latent semantic analysis (LSA) that uses SVD was described by Deerwester et al [2]. Modifying LSA with

a probabilistic approach, based on a multinomial model, Hofmann [3] proposed probabilistic latent semantic analysis (PLSA). Lee and Seung [5] proposed a dimensionality reduction method called nonnegative matrix factorization (NMF) as a topic model that maintains nonnegativity property of data. NMF is also the technique that we use as the basis of our model in this thesis. Last but not least Blei et al. [4] introduced a generative model called latent Dirichlet allocation (LDA) which is the most popular topic model currently in use with NMF. LSA and NMF are decomposition-based methods. PLSA lies between being a decomposition-based method and a generative one. LDA is a generative model. Most other topic modeling approaches can be classified as the extensions of these methods and we will briefly go over some of the related ones to our model.

2.2. Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) is a group of algorithms where a matrix is factorized into two low-rank matrices with the property that all three matrices have no negative elements. This nonnegativity constraint makes the resulting matrices much easier to analyze [5].

Given a nonegative matrix X, NMF seeks the nonnegative matrix factors W and H such that:

$$X \approx WH. \tag{2.1}$$

NMF can be applied to topic modeling domain in the following manner: Given a set of m-dimensional data vectors, the vectors are placed in the rows of an $n \times m$ matrix X where n is the number of documents in the data set. This matrix is then approximately factorized into an $n \times p$ matrix W and a $p \times m$ matrix H. For NMF to generate factors with reduced dimensions compared to the original matrix, p is chosen smaller than n or m.

It is important to understand the approximation in Equation 2.1. We can rewrite this equation row by row as $x^T \approx w^T H$, where x^T and w^T are the corresponding rows of X and W. Each data vector x^T is approximated by a linear combination of the rows of H, weighted by the components of w^T. So H can be regarded as having the basis vectors for the linear approximation of X. In other words, we can now reconstruct a document (row vector) from our input matrix X by a linear combination of our features (row vectors in H) where each feature is weighted by the feature's cell value from the document's row in W (see Figure 2.1).

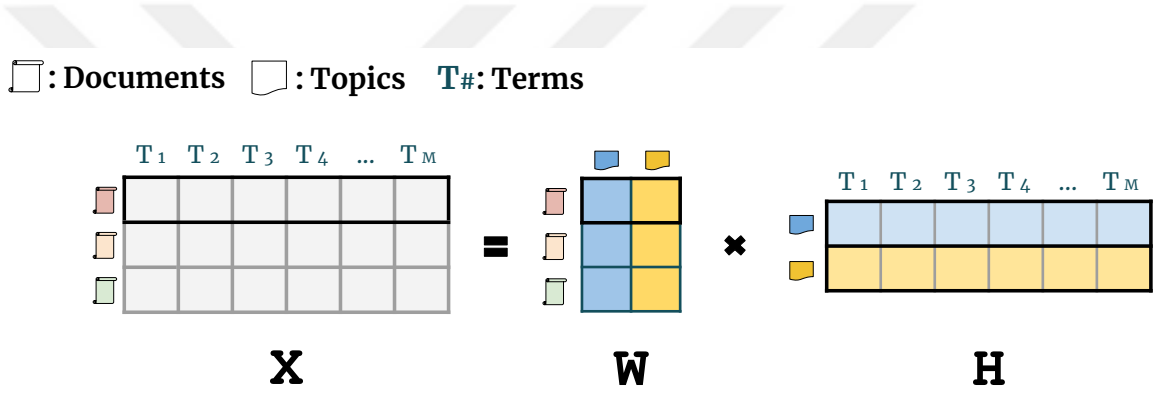


Figure 2.1. Illustration of NMF in terms of linear approximation of X

In topic modeling, W and H have distinctive interpretations where $W_{\nu,i}$ shows the relevance of topic i for document ν and $H_{i,\tau}$ shows the relevance of term τ in topic i. It is also common to call W coefficient (or activation) matrix and H feature (or basis) matrix. It is useful to think of each topic (row vector) in the feature matrix H as a document archetype comprising a set of terms where each term's cell value defines the term's rank in the topic: The higher a term's cell value the higher the term's rank in the topic. A row in the coefficient matrix W represents an original document with a cell value defining the rank of a topic for the document.

As a side note, it is more common to see a column-wise approach for the interpretation of NMF where W is the basis matrix and H is the feature matrix. Instead of rewriting the Equation 2.1 row by row as $x^T \approx w^T H$, it can be rewritten column by

column as $x \approx Wh$, where x and h are the corresponding columns of X and H. This difference comes from using the document-term layout instead of the term-document layout for our data matrix X. There are two main reasons for us to adopt the row-wise approach:

(i) We think that, as a human, it is easier to understand a data matrix that has documents on the rows and terms on the columns like our interpretation of X. A different example than document-term relation can be movie-rating relation. Here again, it is more appropriate to place data vectors into the rows. So in our opinion document-term layout is a more proper and easier to understand layout for the topic modeling.

(ii) scikit-learn [19] is one of the most popular ML library in Python. Its NMF implementation also uses a document-term layout for its data matrix. We made us of scikit-learn's NMF implementation in the source code of this thesis. To make the thesis document and its code consistent and easy to follow we adopted the document-term layout.

Now let's see how to find W and H. The matrices W and H are estimated by minimizing the following objective function:

$$(W, H) = \arg\min_{W,H} D(X \parallel WH), \quad \text{subject to } W, H \geqslant 0 \tag{2.2}$$

where the function D is a suitably chosen cost function. There are several ways in which the W and H may be found with different error functions. Lee and Seung's multiplicative update rule [20] has been a popular method due to the simplicity of implementation and fast convergence. They show the multiplicative update rules for two different cost functions: Square of the Euclidean distance and a form of Kullback-Leibler (KL) divergence.

2.2.1. Cost Function: Square of the Euclidean Distance

The square of the Euclidean distance between two matrices A and B is defined as follows:

$$\|A - B\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2 . \tag{2.3}$$

Euclidean distance is lower bounded by zero and becomes zero if and only if $A = B$. Now we can show the multiplicative update rules for W and H under the Euclidean distance as the cost function:

$$H \leftarrow H \circ \frac{W^T X}{W^T W H} \tag{2.4}$$

$$W \leftarrow W \circ \frac{X H^T}{W H H^T}. \tag{2.5}$$

where $\circ$ denotes the Hadamard (element-by-element) product. It is important to note that the update rules are done on an element by element basis, not matrix multiplication.

2.2.2. Cost Function: Kullback-Leibler divergence

Another useful measure between two matrices A and B is defined as follows:

$$D(A \parallel B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right). \tag{2.6}$$

Like Euclidean distance, this is also lower bounded by zero and becomes zero if and only if $A = B$. However, contrary to Euclidean distance this measure is not a distance but a divergence, because it is not symmetric in A and B. It reduces to KL divergence when $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$, which can be considered as A and B being normalized probability distributions. Now we can show the multiplicative update rules for W and

H under the KL divergence as the cost function:

$$H \leftarrow H \circ \frac{W^T \frac{X}{WH}}{W^T 1} \tag{2.7}$$

$$W \leftarrow W \circ \frac{\frac{X}{WH} H^T}{1 H^T}. \tag{2.8}$$

A detailed and very well explained derivation for both of these multiplicative update rules can be found in the paper written by Burred [21].

2.3. Latent Semantic Analysis

Latent semantic analysis (LSA) is an unsupervised learning technique in NLP. LSA is based on the distributional hypothesis which states that "words which are similar in meaning occur in similar contexts" [22]. LSA consists of three main steps and the second step, dimensionality reduction, is the key part:

Creation of Document-Term Matrix: Like we have mentioned in topic modeling, we first create our document-term matrix that contains term counts per document. We treat documents like bag of words. We can also use metrics other than raw term frequency to quantify the relevance of the terms to the documents.

Dimensionality Reduction: The document-term matrix is decomposed into the product of 3 matrices ($U\Sigma V$) by using SVD. The U matrix corresponds to the document-topic matrix and the V matrix corresponds to the topic-term matrix. Σ matrix is guaranteed to be a diagonal matrix and the diagonal entries of Σ are known as the singular values of our main document-term matrix. The number of non-zero singular values is equal to the rank of our document-term matrix and LSA considers them as potential topics found in the documents. If we keep largest p singular values of Σ with the corresponding p columns of the document-topic matrix U and p rows of the topic-term matrix V, we end up with the best p-dimensional approximation to the original matrix, in the least squares sense [23]. In terms of topic modeling, it means

that we obtained the p most prominent topics found in our original document-term matrix. This is called truncated SVD since it does not keep all of the singular values of the original matrix.

Evaluation: We use the vectors that make up the U and V matrices with different methods to assess the quality of topic assignments to the documents or term distributions of topics or document similarities. The most common method is to use cosine similarity to evaluate similarities between documents or terms. They are compared by taking the cosine angle between any two vectors taken from U or V matrices, respectively.

2.4. Probabilistic Latent Semantic Analysis

Modifying LSA with a probabilistic approach based on a multinomial model, Hofmann proposed Probabilistic latent semantic analysis (PLSA) [3]. PLSA adopts a probabilistic approach instead of SVD (like LSA does) to tackle the topic modeling problem. PLSA models the probability of each co-occurrence $P(D,T)$ of documents D and terms T as a mixture of conditionally independent multinomial distributions:

$$P(D,T) = P(D)\sum_{Z} P(Z|D)P(T|Z) \tag{2.9}$$

$$P(D,T) = \sum_{Z} P(Z)P(D|Z)P(T|Z) \tag{2.10}$$

with Z being the topics. The Equation 2.9 is the asymmetric formulation of PLSA and it can be seen as the probabilistic spin-off of our basic topic modeling assumptions about mixtures. For each document D, a topic Z is drawn from the document's topic distribution, $P(Z|D)$, and a term T is drawn from the term distribution of the topic, $P(T|Z)$. $P(D)$ can be directly determined from the corpus. Essentially the equation tells us how likely to see the document D, and then how likely to find the term T in that document D according to its topic distribution. You can see the plate notation of the asymmetric formulation in Figure 2.2.

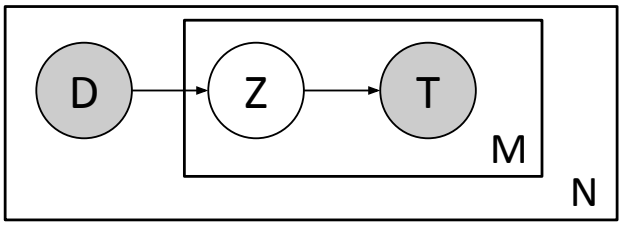


Figure 2.2. Plate notation for the PLSA model (asymmetric formulation). The document D and the term T are observable variables, the topic Z is a latent variable.

The parameters are learned using the Expectation-Maximization (EM) algorithm [3]. EM is an iterative method for finding the likeliest estimates of parameters where the model depends on unobserved latent variables (topics in our case). The number of topics is a hyperparameter that must be chosen in advance and is not estimated from the data.

The Equation 2.10 is the symmetric formulation of PLSA. If we say the asymmetric formulation has the document perspective, then the symmetric formulation has the topic perspective. Here we start with a topic using $P(Z)$, then generate both document D and term T independently from that topic with $P(D|Z)$ and $P(T|Z)$, respectively. You can see the simple illustrations for both the asymmetric and symmetric generation processes in Figure 2.3.

The significance of the symmetric formulation is its resemblance with the LSA formulation. The probability of the topics $P(Z)$ corresponds to the diagonal matrix of singular topic probabilities, the probability of documents given the topic $P(D|Z)$ corresponds to the document-topic matrix U, and the probability of terms given the topic $P(T|Z)$ corresponds to our topic-term matrix V.

PLSA becomes much more flexible than LSA with the probabilistic treatment of topics and terms. But it still has a particular shortcoming: It is not a proper generative model for new documents because it doesn't have a parameter for $P(D)$.

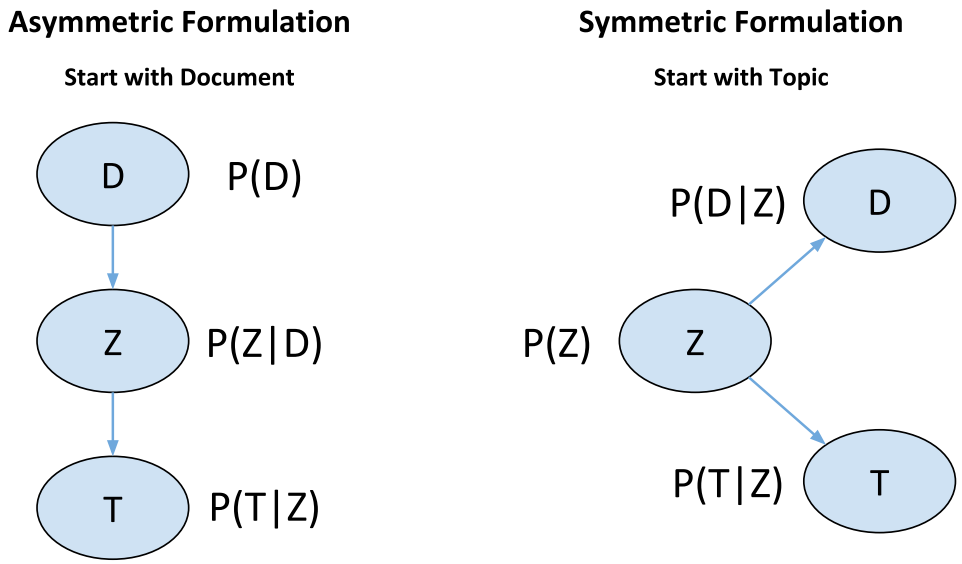


Figure 2.3. Illustrations of asymmetric and symmetric formulation processes of PLSA.

2.5. Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) [4] is the Bayesian version of PLSA where LDA assumes that the topic distribution for a document and the term distribution for a topic have sparse Dirichlet prior. That assumption of Dirichlet prior for documents and topics is what makes LDA a real generative model.

LDA assumes documents are generated in the following fashion: For each document

(i) Choose the number of terms M the document will have.

(ii) Choose a topic mixture for the document. As an example, assume that we have the topics that we gave as an example in Section 2.1: music and food. So, one might choose the document to consist of 60% music and 40% food.

(iii) Generate each term τ in the document by:

  (a) Pick a topic according to the multinomial distribution that we sampled above; we might get the music topic with 60% probability and the food topic with 40% probability.

(b) Use the selected topic to generate the term itself according to the topic's multinomial distribution of terms; if we selected the food topic, we might generate the word "tomato" with 30% probability, "fruit" with 20% probability, and so on.

LDA then backtracks this generative process to find a set of topics that are likely to have generated the document collection.

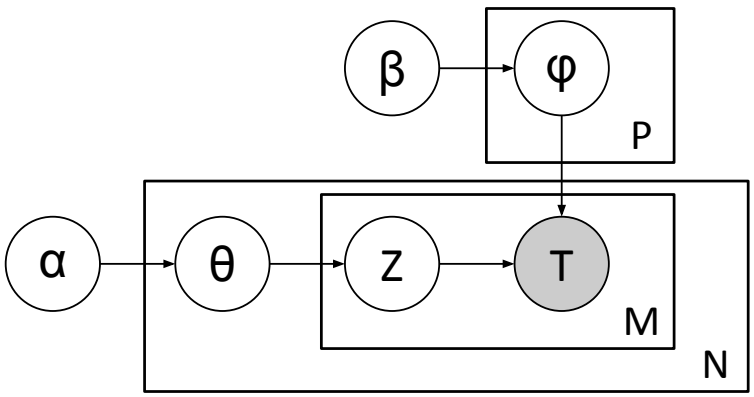


Figure 2.4. Plate notation for the LDA model.

Figure 2.4 is the plate notation representing the LDA model. The variable names are defined as follows:

- P denotes the number of topics
- N denotes the number of documents
- M is the number of terms in a given document
- α is the parameter of the Dirichlet prior on the per-document topic distribution
- β is the parameter of the Dirichlet prior on the per-topic term distribution
- θ_i is the topic distribution for document i
- φ_p is the term distribution for topic p
- z_{ij} of Z is the topic for the j^{th} term in document i
- t_{ij} of T is the specific term.

T being grayed out means that the terms are the only observable variables. α and β are the two hyperparameters that control document and topic similarities, respectively. A low value of α will assign fewer topics to each document whereas a high value of α will assign more. A low value of β will use fewer terms to model a topic whereas a high value will use more terms, thus making topics more similar between each other. The most widely applied variant of LDA today uses sparse Dirichlet priors as suggested in the original paper [4]. The sparse Dirichlet priors encode the intuition that documents cover only a small set of topics and that topics use only a small set of terms frequently. It is also helpful to think θ and φ as the matrices created by the decomposition of the original document-term matrix as we demonstrated with NMF. θ corresponds to the document-topic matrix and φ corresponds to the topic-term matrix.

For LDA, the posterior is intractable to compute and an approximation must be applied. Modern approximate posterior inference algorithms fall into two categories: sampling approaches that are based on Markov chain Monte Carlo (MCMC) sampling and optimization approaches that are based on variational inference. The original LDA paper uses a variational Bayes (VB) algorithm which is the Bayesian version of the variational inference [4]. An alternative inference technique is Gibbs sampling which is a MCMC algorithm [24]. VB is shown to be much more favorable against Gibbs sampling because of the faster convergence and similar accuracy [25, 26].

2.6. Relations Between NMF, PLSA, and LDA

NMF with the objective function as generalized KL divergence can be shown to be equivalent to PLSA. The two approaches only differ in how inference proceeds, but the underlying model is the same [27, 28]. On the other hand, LDA has been developed to address an often-criticized shortcoming of PLSA, namely that it is not a proper generative model for new documents. LDA adds a sparse Dirichlet prior on the per-document topic distribution on top of PLSA. So, PLSA simply becomes the special case of LDA where we assume the Dirichlet prior in the data generating process for LDA is uniform [29].

2.7. Related Extensions

In this section, we will go over some extensions to NMF and LDA that are related to our semi-supervised topic modeling approach.

MacMillan et al. [13] proposed a semi-supervised NMF method called Topic Supervised NMF which can be regarded as one of the most similar studies to ours in the literature. They use the square of the Euclidean distance as the cost function and multiplicative update as the update rule. They first create a supervision matrix that shows which topics are permitted in which documents and then they apply the update rules with the supervision matrix. This structure enables the model to learn topics from specific documents. To measure the success, they calculate a similarity matrix that expresses the Jaccard distance between the identified and true document-topic matrices.

A semi-supervised NMF model proposed by Lee et al. [14] tries a joint factorization of the data matrix and the label matrix which holds topic labels of documents. They consider two tasks in their experiments:

- *Supervised feature extraction for classification:* Split data as train and test. Obtain feature matrices using the semi-supervised NMF model then train a classifier with these feature matrices. Lastly, show the results in terms of classification accuracy.
- *Semi-supervised clustering:* Apply k-means on the feature matrix learned by the semi-supervised NMF model and the traditional NMF to see the differences. As the output, they plot clustering accuracy to the ratio of known labels which goes from 0 (unsupervised) to 100 (fully supervised).

Wang et al. [15] proposed a semi-supervised NMF method with pairwise constraints that propagates both the must-link and cannot-link constraints from the constrained samples to unconstrained samples. Using this information, they adjust the

data weight matrix and apply it as a regularization term to the NMF objective function.

Another constraint-based semi-supervised NMF method that focuses on multi-label learning is proposed by Liu et al. [16]. They define input-based similarity and class-based similarity which correspond to the similarity between input data and the similarity of labels assigned to these data, respectively. They build the model upon the assumption of if the label assignments of the documents were logical, then the similarities defined above should be also consistent with each other.

Several semi-supervised data clustering methods using NMF are also trying to achieve a similar goal as ours. The common approach is to use two types of pairwise constraints on data objects: must-link (the two data points must belong to the same class) and cannot-link (the two data points cannot belong to the same class). Chen et al. [17,30] perform trifactorizations on the data matrices with a newly learned distance metric based on the pairwise constraints to improve the quality of clustering. On top of must and cannot link constraints, Wang et al. [31] use inter-type and intra-type relationship constraints to further guide the clustering algorithm.

Blei et al. [32] showed a successful hierarchical topic model with hierarchical latent Dirichlet allocation (hLDA). However because this model is unsupervised, it can't make use of any information from hierarchical labels. Petinot et al. [33] proposed a supervised version of hLDA called hierarchical labeled latent Dirichlet allocation (hLLDA) to be able to use the hierarchical label information to automatically generate corresponding topics. But being a supervised model, hLLDA cannot discover new latent topics. To tackle all the shortcomings of both model, Mao et al. [34] proposed a semi-supervised hierarchical topic model that is the generalized version of both hLDA and hLLDA called semi-supervised hierarchical latent Dirichlet allocation. The model makes use of the hierarchical labels and can discover new latent topics.

Another supervised LDA model proposed by Blei et al. [35] focuses on prediction problems and handles them like a regression model. They add a response variable to LDA associated with each document and jointly model the responses and documents to discover latent topics. They compare their supervised LDA with a modern regularized regression and a traditional unsupervised LDA followed by a separate regression.

Labeled LDA proposed by Ramage et al. [36] claims to improve previous supervised LDA models by enabling each document to have more than one label (topic). They focus on the credit attribution task which is matching parts of a document with the most appropriate labels. To achieve this, labeled LDA constrains the traditional LDA by defining a one-to-one correspondence between the latent topics learned by the model and user labels.

3. THEME SUPERVISED NONNEGATIVE MATRIX FACTORIZATION

Our proposed *Theme Supervised Nonnegative Matrix Factorization (TSNMF)* model is a semi-supervised variation of the traditional NMF. Instead of giving the final model directly, we will build and explain our model step by step starting from traditional the NMF. We will demonstrate all the steps on a particular example that will evolve with each step. For the examples; suppose we have 5 documents with a dictionary of M terms. The number of topics will be two in each step. The sentences that represent the documents are as follows:

- **Document 1:** Pineapple pizza is the best.
- **Document 2:** Eggplant is an underrated vegetable.
- **Document 3:** Brontosaurus are like ancient giraffes.
- **Document 4:** Baby shark is safe at last.
- **Document 5:** There is an owl in the chimney of my house.

Note 1: For the following figures, a cell's value is greater than or equal to zero if it is not specifically indicated.

Note 2: We use percentages to show topic distributions of documents and term distributions of topics in demonstrations for easier understanding. However, the resulting matrices from decompositions don't need to include proper distributions. We discuss these issues in Chapter 5 and Chapter 6 in detail.

3.1. Step 1: Traditional Nonnegative Matrix Factorization

Given a set of m-dimensional data vectors, the vectors are placed in the rows of an $n \times m$ matrix X where n is the number of documents. The data is generally represented as bag-of-words with either their term frequency (tf) or term frequency–inverse docu-

ment frequency (tf-idf) values. The tf, the number of times a term occurs in a given document, is multiplied with idf component to calculate tf-idf, which are computed as follows:

$$idf\,(\tau) = \log \frac{1+n}{1+df\,(\tau)} + 1 \tag{3.1}$$

$$tf\text{-}idf\,(\tau, d) = tf\,(\tau, d) \times idf\,(\tau) \tag{3.2}$$

where n is the total number of documents in the document set, and $df(\tau)$ is the number of documents in the document set that contain term τ. This matrix is then approximately factorized into an $n \times p$ matrix W and a $p \times m$ matrix H. For NMF to generate factors with reduced dimensions compared to the original matrix, p is chosen smaller than n or m.

In topic modeling, W and H have distinctive interpretations where $W_{\nu,i}$ shows the relevance of topic i for document ν and $H_{i,\tau}$ shows the relevance of term τ in topic i. It is also common to call W coefficient (or activation) matrix and H feature (or basis) matrix. It is useful to think of each topic (row vector) in the feature matrix H as a document archetype comprising a set of terms where each term's cell value defines the term's rank in the topic: The higher a term's cell value the higher the term's rank in the topic. A row in the coefficient matrix W represents an original document with a cell value defining the rank of a topic for the document. The matrices W and H are estimated by minimizing the following objective function:

$$(W, H) = \arg\min_{W,H} D(X \parallel WH), \quad \text{subject to } W, H \geqslant 0 \tag{3.3}$$

where the function D is a suitably chosen cost function. We use both square of the Euclidean distance and KL divergence as the cost function in our experiments. We use the multiplicative update rule to find the best W and H.

3.1.1. Demonstration

You can see the demonstration of traditional NMF with 2 topics In Figure 3.1. There is no supervision or prior information in this model. Let's analyze the decomposition closer with the example that we gave at the beginning of this chapter.

- To understand what topics represent, we need to analyze topic-term matrix H:
  - **Topic A:** 20% pizza, 15% best, 15% eggplant, 5% giraffe, 5% pineapple, ... (at which point, we could interpret topic A to be about **food**)
  - **Topic B:** 25% shark, 20% owl, 10% like, 5% vegetable, 5% brontosaurus, ... (at which point, we could interpret topic B to be about **animals**)
- In document-topic matrix, W, it is highly likely to see topic distributions like this:
  - **Documents 1 and 2:** 85% Topic A, 15% Topic B
  - **Documents 3, 4 and 5:** 20% Topic A, 80% Topic B

Even in this simple example, we needed to analyze both matrices simultaneously to understand the general structure of the document set. There is a lot of room for ambiguity caused by the terms that can belong to multiple topics.

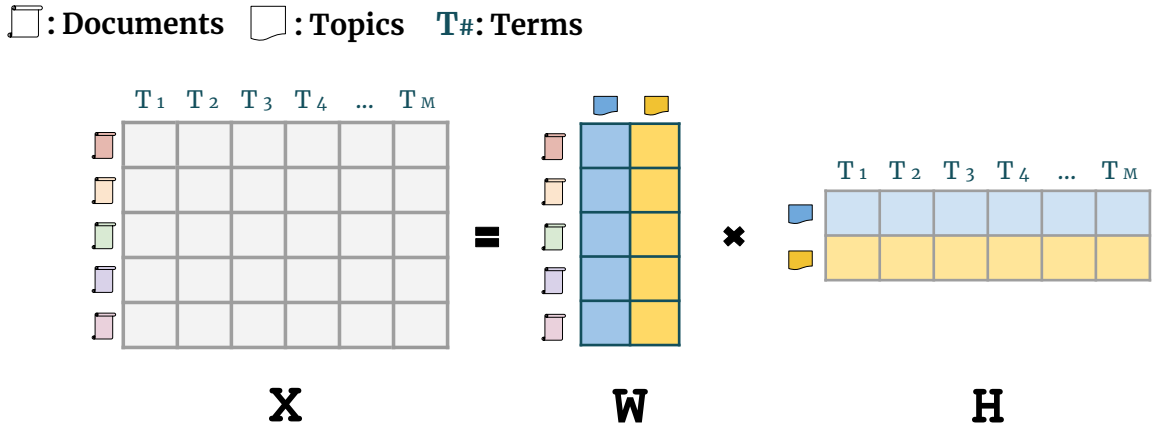


Figure 3.1. Demonstration of Step 1: Traditional Nonnegative Matrix Factorization

3.2. Step 2: Supervision

Supervision enables the user to provide documents with labels and constrains the representation of the topics to align with these labeled documents. Supervision can be represented by the $n \times p$ topic supervision matrix S. The elements of S are of the following form:

$$S_{\nu,i} = \begin{cases} 1 & \text{if topic } i \text{ is permitted in document } \nu \\ 0 & \text{if topic } i \text{ is } \textit{not} \text{ permitted in document } \nu. \end{cases} \tag{3.4}$$

We use the supervision matrix S to constrain the coefficient matrix W. For all pairs (ν, i) such that $S_{\nu,i} = 0$, we enforce that $W_{\nu,i}$ must also be 0. This constraint can be viewed from two different perspectives. First, the labeled documents must lie in the subspace of their assigned topics. Second, the topics must adapt to the documents they are permitted to. The supervision matrix also allows documents to belong to multiple topics.

Let $\circ$ denotes the Hadamard (element-by-element) product. For a term-document matrix X and supervision matrix S, the new NMF with supervision seeks matrices W and H that minimize:

$$(W, H) = \arg\min_{W,H} D(X \parallel (W \circ S)H), \quad \text{subject to } W, H \geqslant 0. \tag{3.5}$$

The change in the objective function does not affect the derivation of the multiplicative update rules. We only need to replace W with $W \circ S$ in the updates.

Multiplicative update with square of Euclidean distance as the cost function:

$$H \leftarrow H \circ \frac{(W \circ S)^T X}{(W \circ S)^T (W \circ S) H} \tag{3.6}$$

$$W \leftarrow W \circ \frac{(X H^T) \circ S)}{((W \circ S) H H^T) \circ S}. \tag{3.7}$$

Multiplicative update with KL divergence as the cost function:

$$H \leftarrow H \circ \frac{(W \circ S)^T \frac{X}{(W \circ S) H}}{(W \circ S)^T 1} \tag{3.8}$$

$$W \leftarrow W \circ \frac{\left(\frac{X}{(W \circ S) H} H^T\right) \circ S)}{(1 H^T) \circ S}. \tag{3.9}$$

3.2.1. Demonstration

Before training the model, the topics of Documents 1, 2, 3, and 4 were given to us. Documents 1 and 2 are about food. Documents 3 and 4 are about animals. We accepted Topic A as the food and Topic B as the animal topics (can also be the opposite). We permit Topic A in Documents 1 and 2 but not Topic B, vice versa for Documents 3 and 4. Because the theme of Document 5 is unknown, we permit both Topic A and B in Document 5. You can see the demonstration of NMF with supervision in Figure 3.2. When we check the Document 5, we can see that it is more similar to Documents 1 and 2 than Documents 3 and 4. In other words, Document 5 is more about animals than food. Thus, we expect to see higher values in its Topic A score than Topic B score.

When we analyze the decomposition closer, the main difference from the unsupervised version is that topic distributions of labeled documents are preset - they have to be in the latent subspace of the topics they are assigned to. (The only uncertainty can occur when a document has multiple labels where the labels can get different weights.)

We get two major benefits out of this approach: First, better term distributions for topics. Second, unlabeled documents can be categorized into topics that we chose via labeled documents. We no longer need to rely on topic models to find meaningful topics. We showed the second benefit in the introduction part via the zodiac sign example and in this demonstration, we emphasized the first benefit.

- We already know what topics represent, but with supervision we get more precise topic representations where more specific terms get higher percentages:
  - **Topic A (food):** 25% pizza, 20% eggplant, 10% pineapple, 5% best, ...
  - **Topic B (animal):** 30% shark, 20% owl, 10% giraffe, 5% brontosaurus, ...
- We check the document-topic matrix, W, to see the topic distribution of Document 5:
  - **Documents 1 and 2:** 100% Topic A, 0% Topic B
  - **Documents 3 and 4:** 0% Topic A, 100% Topic B
  - **Document 5:** 4% Topic A, 96% Topic B

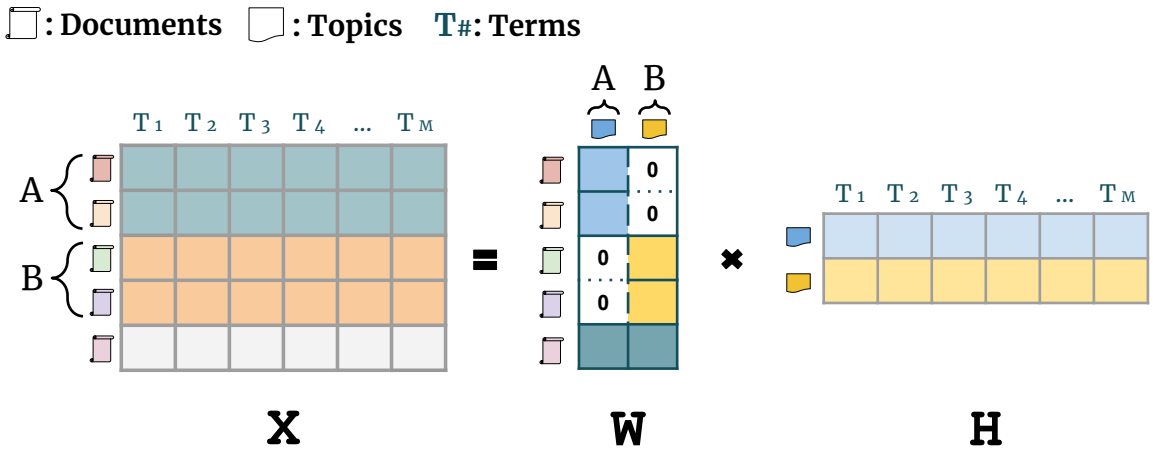


Figure 3.2. Demonstration of Step 2: Supervision

3.3. Step 3: Themes and Subtopics

In traditional NMF, there need not be a direct connection between topics. Every topic tries to cover different aspects of the document set and some topics can hold more information than others. For example, there could be a topic about music which includes metal and classical music at the same time. There could also be another topic just about cats. If we increase the topic count, then we may get 2 separate topics about classical music and metal music plus the cats topic. But actually, we can not be sure about what will be the contents of the new topics when we increase the topic count in the unsupervised setting. This situation is also more interesting for the case with the supervision because we increase the topic count to discover new topics, but at the same time, supervision forces a restriction on the topics.

We propose a new hierarchical topic structure that consists of themes and subtopics. Themes are a direct replacement of topics; but unlike topics, a theme consists of a collection of subtopics. Every theme has the same number of subtopics. If we set the number of subtopics to one, then the model becomes the regular topic model where each theme is represented by one subtopic. Instead of permitting topics in the documents, we permit themes and their subtopics in the documents. This structure allows a better interpretation and representation of topics and documents. Subtopics are not predetermined and we only know the themes that subtopics are connected to. So, each subtopic can reflect different aspects of a theme.

With themes, we can now safely take the music topic as a theme and train the model with two subtopics to obtain two different implications of music: Classical and metal. The concept of themes and subtopics is useful when it is used with the supervision which helps to better understand the provided topics.

3.3.1. Demonstration

Now food and animal are not topics, but themes. Theme A is the food theme and Theme B is the animal theme. Both themes have 3 subtopics. We permit Theme A and its three subtopics in Documents 1 and 2 but not Theme B, vice versa for Documents 3 and 4. Because the theme of Document 5 is unknown, we permit both Theme A and B in Document 5. You can see the demonstration of NMF with supervision, themes, and subtopics in Figure 3.3.

Also before themes, analyzing the topic distribution of the labeled documents was redundant. But now with themes and subtopics, we can analyze how labeled documents that belong to the same theme differentiate from each other and what their specialized subtopics are.

- To understand what subtopics represent, we need to analyze topic-term matrix H (subtopic names next to arrows are the most likely predictions based on the term distributions):
  - **Subtopic A_1:** 70% pizza, 15% pineapple, 10% pasta, ... $\rightarrow$ **Italian Food**
  - **Subtopic A_2:** 50% pineapple, 30% apple, 5% eggplant, ... $\rightarrow$ **Fruits**
  - **Subtopic A_3:** 60% vegetable, 25% eggplant, 10% onion, ... $\rightarrow$ **Vegetables**
  - **Subtopic B_1:** 50% brontosaurus, 40% t-rex, 5% giraffe, ... $\rightarrow$ **Dinosaurs**
  - **Subtopic B_2:** 70% baby shark, 20% kitten, 5% safe, ... $\rightarrow$ **Cute Animals**
  - **Subtopic B_3:** 55% owl, 30% canary, 10% chimney, ... $\rightarrow$ **Birds**
- We check the document-topic matrix W, to see the subtopic distributions of the documnets:
  - **Document 1:** 70% Subtopic A_1, 25% Subtopic A_2, 5% Subtopic A_3
  - **Document 2:** 5% Subtopic A_1, 10% Subtopic A_2, 85% Subtopic A_3
  - **Document 3:** 85% Subtopic B_1, 10% Subtopic B_2, 5% Subtopic B_3
  - **Document 4:** 10% Subtopic B_1, 90% Subtopic B_2, 0% Subtopic B_3
  - **Document 5:** 4% Theme A; 6% Subtopic B_1, 20% Subtopic B_2, 70% Subtopic B_3

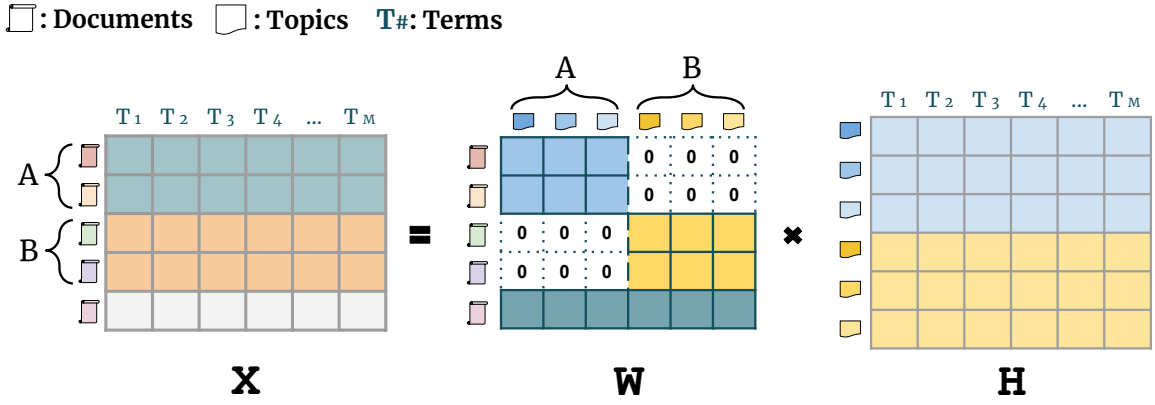


Figure 3.3. Demonstration of Step 3: Themes and Subtopics

3.4. Step 4: Background Topic

In NLP, cleaning the common and unnecessary terms out of the documents has always been an occurring issue. There are NLP methods like removing stopwords or removing terms that have less than some number of letters. These methods are useful but they don't use any context from the documents.

We introduce a static topic called *background topic* to the model that is permitted in every document. Unlike themes, background topic learns from every document. The background topic tries to generalize things that are common for every document. By doing that, the background topic removes the common and generally unwanted information from the environment; and helps themes to carry more exclusive meanings. In other words, themes can focus on more important terms in the documents to which they are assigned, thanks to the background topic. Also background topic is not a theme because there is no supervision and we don't need a hierarchical structure for it. With the addition of the background topic, we have completed the first version of our proposed model and it is called *Combined Theme Supervised Nonnegative Matrix Factorization*.

Note: Throughout this thesis, even though *TSNMF* uses themes, subtopics, and background topics instead of topics, we will continue to use the term topic as a concept that encompasses all of these structures.

3.4.1. Demonstration

Here we added the background topic to the previous model. Background topic is permitted in every document. You can see the demonstration of *Combined TSNMF* in Figure 3.4. As explained above, the background topic boosts the importance of theme-specific terms by collecting common terms on itself. An example term distribution for background topic can be as follows: 10% all, 8% use, 6% like, 5% time, ...

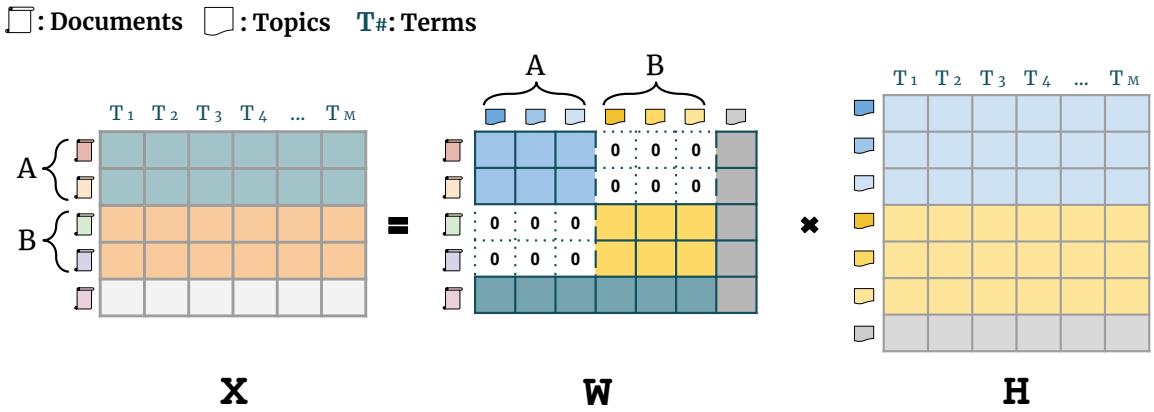


Figure 3.4. Demonstration of Step 4: Background Topic

3.5. Step 5: Separation

The last step is the separation of themes. The motivation is imitating the one vs all training procedure. Instead of training all the themes and the background topic together, we train a different model for each theme with a background topic. The binding part between the models is the background topics because the background topics are permitted in every document and every model has the same background topic structure. As we have mentioned, the procedure is like one-vs-all: theme-vs-background.

This final version of our proposed model is called *Separated Theme Supervised Non-negative Matrix Factorization* and we will use this model while explaining methods in the upcoming chapters.

3.5.1. Demonstration

In the *Separated TSNMF*, we train a separate model for each theme. Instead of one distribution over all subtopics in one model, we get separate subtopic distributions with a background topic for each theme. To understand which theme a document belongs to, we first calculate a score for each theme by comparing subtopic values to background topic value in each W. We expect higher values in subtopic values relative to background topic value if the document belongs to this theme. Then we compare those calculated scores to find the theme of documents. You can see the demonstration of *Separated TSNMF* in Figure 3.5. Let's have a more detailed look at W_1 and W_2 matrices for Document 5:

- **W_1 (Theme A - Food):** 2% Subtopic A_1, 5% Subtopic A_2, 3% Subtopic A_3, 90% Background Topic
- **W_2 (Theme B - Animals):** 5% Subtopic B_1, 17% Subtopic B_2, 75% Subtopic B_3, 3% Background Topic

Because Document 5 is about animals and more specifically birds, subtopic scores in W_2 are higher than the background topic score, while in W_1, background topic score is higher than the subtopic scores. More detail about theme scoring for documents will be in Chapter 5.

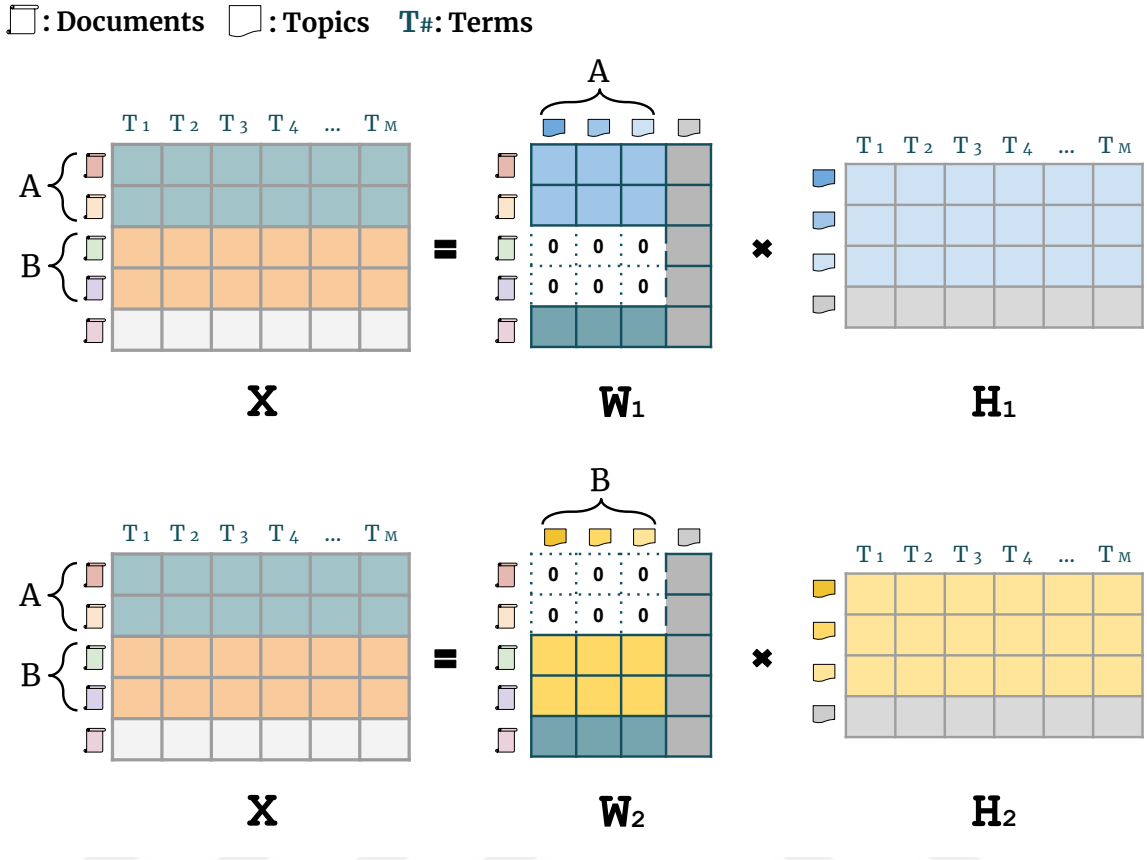


Figure 3.5. Demonstration of Step 5: Separation

3.6. Extension: Fully Supervised

Fully supervised approach can be likened to supervised ML approaches where the data is separated into training and test datasets. Here we take labeled documents as the training dataset and unlabeled documents as the test dataset. To train the model, we only use the labeled documents. The purpose of the training is to obtain theme-term matrices, H (those matrices represent what themes are). We don't need the resulting document-theme matrices, W, because we already know the labels of the documents that we used in training. But one may still want to inspect the resulting document-theme matrices to see the subtopic distribution of the documents to gain more insight about the labeled documents. After the training process, we create a new data matrix (document-term matrix), X, from unlabeled documents. Then we decompose this new X matrix using *TSNMF*, but this time we fix theme-term matrices to the matrices

that we have obtained from the training. With this, we obtain theme distributions of every unlabeled document via document-theme matrices.

This approach restricts us in two ways: First, it restricts us to use only the predefined labels (topics). So, we can not discover any new topics. Second, it restricts the training dataset to consist of labeled documents. So, we can not add terms from the test dataset to the dictionary. But, using only labeled documents in the training process gives us more precise theme-term matrices and better-identified themes.

3.6.1. Demonstration

You can see the demonstration of a fully supervised version of *Separated TSNMF* in Figure 3.6. The first part of the figure demonstrates the training procedure. The only difference from Step 5 is that we only use labeled documents to learn the themes. Thus, Document 5 is not involved in the first part. We store the theme-term matrices, H_1 and H_2, to use in the testing part. In the second part, we first generate our data matrix (X_{test}) from the unlabeled documents. Under normal conditions, *TSNMF* decomposes the data matrix to find both W and H. However, this time we have already obtained our theme-term matrices, H_1 and H_2, in the first part. Thus, we fix H matrices and *TSNMF* finds only document-theme matrices, W_{test_1} and W_{test_2}. Finally, we calculate the *theme scores* of Document 5 using these matrices.

3.7. Extension: New Topics

In this extension, we will show how *TSNMF* can explore new topics from unlabeled documents. To explore new topics, we first need to decide on the number of new topics. Then instead of creating a separate theme for each new topic, we create only one theme and accept new topics as the subtopics of this theme. So, if we anticipate there could be four new topics, we create a theme with four subtopics and a background topic to discover these four new topics. Finally, we permit new topics in only unlabeled documents (and maybe in some labeled documents that may have extra topics but this is a rare case).

3.7.1. Demonstration

In this example, we are searching for 2 new topics and we also add one new unlabeled document to the document set:

- **Document 6:** Classical music is metal music before electricity.

You can see the demonstration of exploring new topics with *Separated TSNMF* in Figure 3.7. The main difference from the original example is that we have a third theme with two subtopics that represents the new topics. The 2 new topics are only permitted in two unlabeled documents. In the examples without new topics, Document 5 was put under Theme B (animal theme), because it is about birds. Here, Document 5 will also get high scores for Theme B. But, one of the new topics (e.g. the first new topic) can also specialize in Document 5 and this new topic can be about *owls*. For Document 6, it belongs to neither Theme A nor Theme B. So we expect the other new topic (e.g. the second new topic) to be about Document 6 which is *music*.

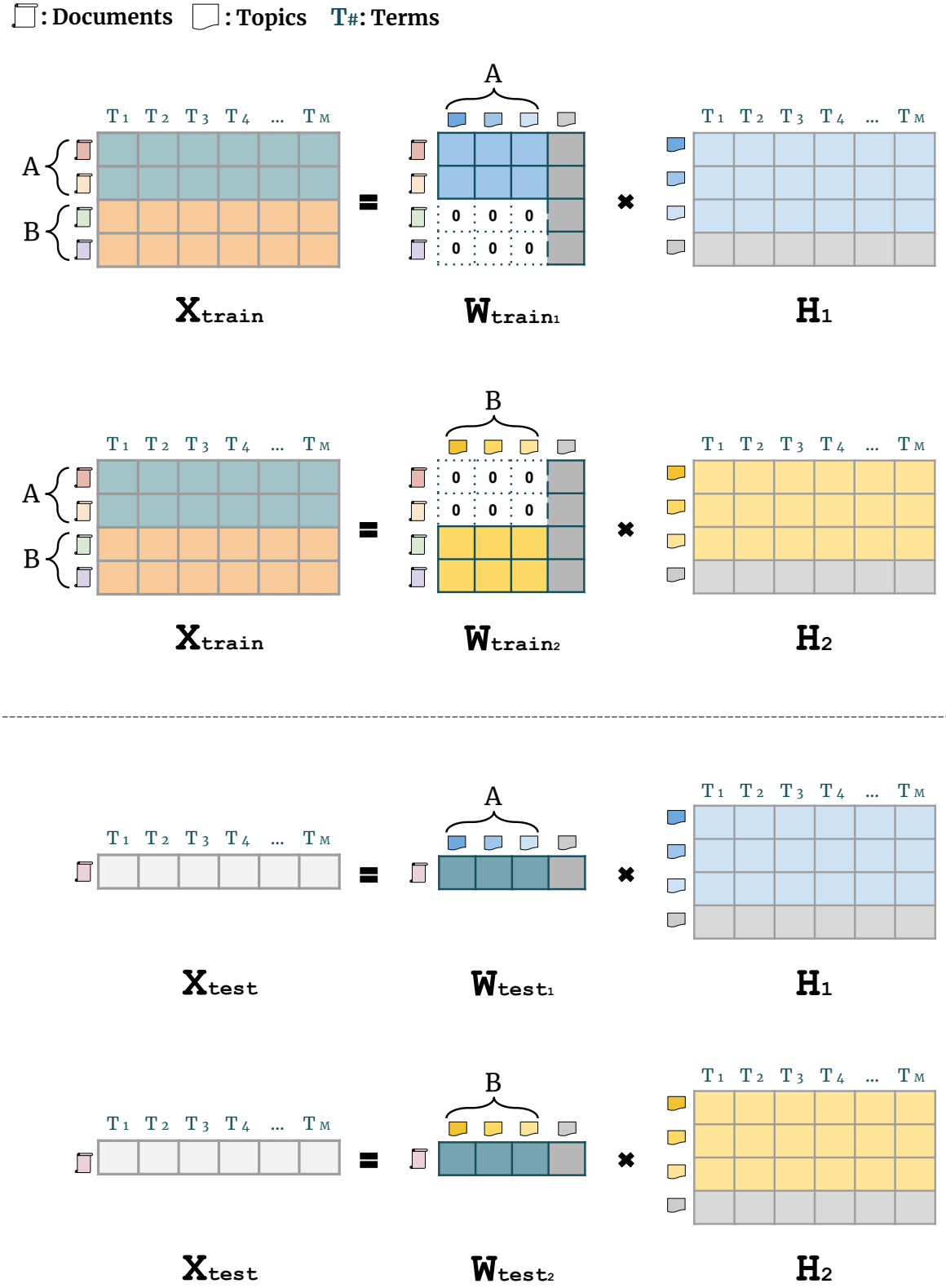


Figure 3.6. Demonstration of the fully supervised version of *TSNMF*

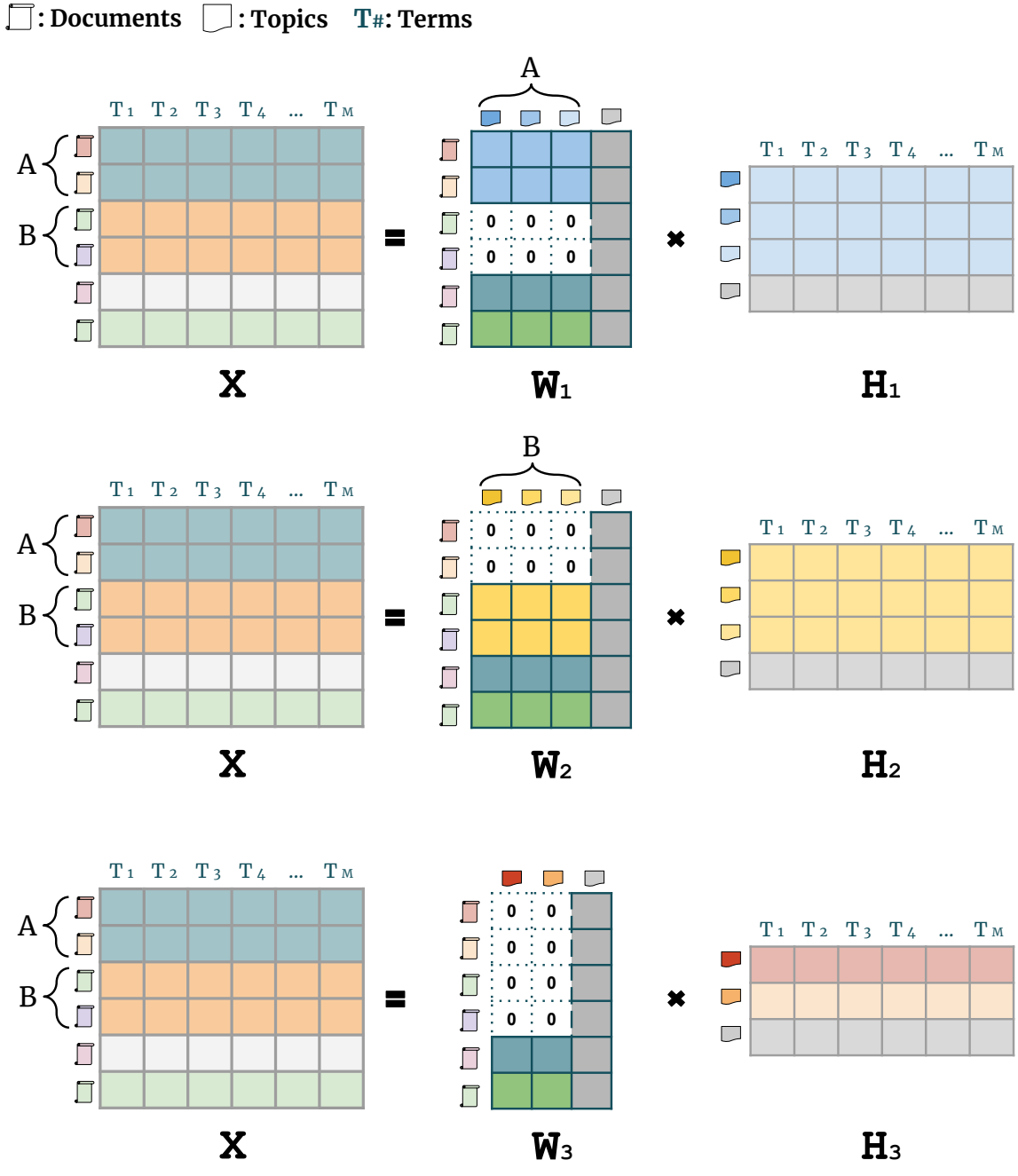


Figure 3.7. Demonstration of exploring new topics with *TSNMF*

4. BCOOL INITIALIZATION

NMF has distinct advantages in matrix factorization: The factorized matrices maintain sparsity and nonnegativity of the data matrix, X, which enables them to be more interpretable [5]. However, NMF has also its disadvantages. The optimization problem for Equation 4.1 has been shown to generalize k-means clustering problem which is known to be NP-complete [37]. The multiplicative update rule for this optimization problem only guarantees to find a local minimum, rather than a global minimum, since it is convex in either W or H, but not both.

$$(W, H) = \arg\min_{W,H} D(X \parallel WH), \quad \text{subject to } W, H \geqslant 0 \tag{4.1}$$

In practice, it is possible to run NMF with different initial setups and choose the one with the best local minimum. However, this reduces the replicability of the solution where even a slight change in parameters may produce different NMF factors. Thus, the initialization of the model is critical to obtain consistent results.

The most simple and preferred initialization method is random initialization where coefficient and feature matrices are initialized as dense matrices of random numbers between 0 and 1. However, since each random initialization may end up at a different local minimum, we propose a very inexpensive and highly consistent semi-deterministic initialization method called *bCool* to initialize the feature matrix H (inspired by the works of Langville et al. [38]).

bCool makes use of the structure of our proposed *TSNMF* model to initialize the topic-term matrix, H. Rather than initializing H with a dense random matrix, it makes more sense to initialize it using the given sparse document-term matrix X. *bCool*

initializes each row of H by averaging specific rows of X. It uses document vectors (row vectors of X) that belong to a specific theme to initialize the theme-related rows of H, and a sample from all documents to initialize background topic related row of H. The steps to initialize H with *bCool* are as follows:

(i) Group documents by their themes. If a document has multiple themes, then add this document to multiple groups.

(ii) Sort each group in descending order according to the documents' densities which is simply the number of unique terms in a document (or typically the row length for sparse matrices).

(iii) Keep the densest half of the documents for each group. Discard the other halves.

(iv) Split each group into N_I subgroups where the cumulative density of each subgroup is approximately equal (N_I is the number of latent topics).

(v) If there is not enough document for each subgroup (number of documents is less than the number of latent topics), then assign documents - from any theme - randomly with direct proportion to their densities. So the denser a document is, the higher probability it will be chosen. The number of documents to be assigned to a subgroup is proportional to the ratio between the size of the training data and the number of themes.

(vi) Calculate average row vectors for each subgroup by taking the mean of corresponding rows of X for each document in the subgroup. Then assign those average row vectors to the related rows of H matrix.

- As an example assume that there is a theme called *food*, there are 90 documents with *food* theme, and the number of subtopics is 3. First, we sort these 90 documents in descending order according to their densities and eliminate the last half. Then, split the remaining 45 documents into 3 (number of subtopics) subgroups where the cumulative density of each subgroup is approximately equal. Calculate 3 separate average row vectors for each subgroup using the data matrix X. Last, if it is a *Separated TSNMF* model, assign those vectors to the first 3 (number of subtopics) rows of the *food* theme's H. If it is *Combined TSNMF* model, assign those row vectors to

food theme-related rows of H.

(vii) For the background topic, take quarter of the densest documents from each group (theme). Calculate the average row vector by taking the mean of all the chosen documents using X matrix, then assign it to the background topic related row of H. So, for the *Separated TSNMF* model, initial values of background topic related row for all H matrices will be the same. For the *Combined TSNMF* model, there is only one H matrix and it has only one row for the background topic.

5. SCORING OF DOCUMENTS

The main objectives of topic modeling are discovering latent topics from a set of documents and clustering those documents under the discovered latent topics. Then it is important to express the connection between documents and topics with comprehensible methods and to find a good way to represent your outcomes. We will deal with the representation of the outcomes, especially topics, in the next chapter (Scoring of Terms). In this chapter, we will focus on document-topic relation and present a novel idea for scoring documents using subtopics and themes.

The document-topic distribution matrix, W, holds the distribution over topics for each document. Each row vector is the topic distribution of a document. However, these distributions don't have to be proper probability distributions where the values don't have to sum up to 1. Higher the value of a cell on a row means the document is more related to the corresponding topic. In traditional NMF, row vectors of the document-topic matrix are used to rank topics for each document. Table 5.1 shows a simple case with 4 documents and 3 topics (the table is just an easier to understand version of a document-topic matrix). While some documents belong exactly to one topic like Document 1 $\leftarrow$ Topic A, some documents have multiple topics like Document 3 $\leftarrow$ Topic B & C and Document 4 $\leftarrow$ Topic A & B. The ranking gives us an easy to understand representation. But using just rankings and omitting the numerical values cause us to lose the significance of topics for each document. On the other hand, if we try to use the values of topic scores directly; because the values range from 0 to +infinity, it is near impossible to make sense of these individual scores. For example; in Table 5.1, when we examine Document 3 and Document 4, even though they both have two topic scores that are close to each other and greater than the other topic score; topic scores of Document 4 are strictly greater than topic scores of Document 3. This situation should be included in the representation. Then we need to find a way that preserves the numerical significance of scores while keeping topic scores more

Table 5.1. Values of an example document-topic matrix.

	Topic A	**Topic B**	**Topic C**
Document 1	100	2	5
Document 2	3	4	3
Document 3	1	45	54
Document 4	25	25	1

understandable for human interpretation.

The most standard method is normalizing W matrix over topics for each document which is the same thing as normalizing each row separately. This is a pretty simple and effective method that converts all the scores to a range between 0 and 1 while conserving their rankings. However, it fails to preserve the numerical significance of the scores because the normalization happens separately for each row. When we normalize the W matrix in Table 5.1, we get the topic scores for each document as shown in the Table 5.2 and as you can see, the difference between the topic scores of Document 3 and Document 4 is not carried to the final scores with standard normalization.

Table 5.2. Normalized values of the example document-topic matrix.

	Topic A	**Topic B**	**Topic C**
Document 1	0.93	0.02	0.05
Document 2	0.30	0.40	0.30
Document 3	0.01	0.45	0.54
Document 4	0.49	0.49	0.02

5.1. Theme Score

Before moving on to our scoring method, it is important to remember that our main *Separated TSNMF* model generates a different document-theme matrix, W^k, for each theme k. If there are 3 subtopics in a theme, then we will have 3 different topic scores (plus a background topic score) for each theme. (As a reminder, we don't count background topics as subtopics of themes, because their main purpose is to regularize subtopics by representing all the documents and themes.) These subtopic scores are useful but eventually, we need one comprehensive score, namely the *theme score*, to represent the importance of the themes for the documents. We can choose the subtopic with the highest value to represent the theme and use its score. The problem with this approach is that we will end up with values from different document-theme matrices which may have different boundaries and such. So it is not reasonable to compare these values to rank themes for documents.

Therefore we need to convert subtopic scores to a more workable format before choosing the highest one, then we can compare scores from other themes. Here we propose our document scoring method that utilizes hierarchical topic structure and background topics of *TSNMF* to overcome foretold problems. We first apply normalization to subtopic scores using the background topic score separately for each theme. This normalization converts subtopic scores to a more comprehensible and comparable format. Then we choose the highest valued normalized subtopic scores to be the *theme scores* of documents. The normalization also enables the comparison of different *theme scores* possible, because it projects all the values to the same boundaries, between 0 and 1. We normalize the subtopic score of subtopic i under theme k for document ν as $\frac{W^k_{\nu,i}}{W^k_{\nu,i}+W^k_{\nu,\neg I}}$. Then the *theme score* of theme k for document ν forms as follows:

$$\textit{theme score} \triangleq \max_i \frac{W^k_{\nu,i}}{W^k_{\nu,i} + W^k_{\nu,\neg I}} \tag{5.1}$$

where $\neg I$ represents the background topic. $W^k_{\nu,i}$ is the raw score of subtopic i under theme k for document ν that shows how important this subtopic is for this document.

$W^k_{\nu,\neg I}$ is the score of the background topic for theme k and document ν. We use background topics as the representation of recurring terms for all the documents. When a document has terms that are specific to a theme k, the subtopic scores for theme k should get higher scores than the background score of theme k. Or at least the subtopic-background ratio for theme k should be higher than other themes' ratios. Here in the *theme score* formulation, we represented this ratio between the subtopic score and the background score of a theme like a normalization term so that all the scores can have values between 0 and 1 and can be easily compared. After normalizing all the subtopic scores, we choose the highest subtopic score for each theme to become its *theme score*. Table 5.3 shows some document scores before and after applying theme scoring, respectively.

The power of this method comes from the background topics' being both a separate local variable for each theme and a global variable by representing all the documents and themes at the same time. However, for the *Combined TSNMF* model, there is only one document-theme matrix, W, and because of that, there is only one background topic score for each document. For that reason, we won't observe that much of an effect of our theme scoring scheme on the *Combined TSNMF* model. Because we are losing the local property of the background topics and theme scoring becomes a simple normalization that divides all the subtopic scores of a document with the same value.

Table 5.3. Top table shows the subtopic scores of some documents before any process. Bottom table shows the subtopic scores after applying theme scoring scheme. Bold scores are the *theme scores* of the documents.

	Theme A			**Theme B**		
	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_B$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_B$
Document 1	100	2	5	3	4	3
Document 2	45	54	1	25	25	1
Document 3	2	1	20	5	8	40

	Theme A			**Theme B**		
	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_B$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_B$
Document 1	**0.95**	0.28	-	0.50	**0.57**	-
Document 2	0.98	**0.98**	-	**0.96**	**0.96**	-
Document 3	**0.09**	0.05	-	0.11	**0.17**	-

6. SCORING OF TERMS

The power of NMF comes from its interpretability. The training of the model forms the topic-term matrix, H, which is a distribution over terms for each latent topic. While it is possible to use the values from H matrix directly to rank the terms; our model *TSNMF* offers much more thanks to its semi-supervised nature and theme-based structure. Using these advantages, we present a new measure called *purity* and a new term scoring scheme called *Theme Term Score (TTS)* which allows more tailored terms for each latent topic and also for each document.

6.1. Purity and Theme Term Score

In the traditional NMF model, topic-term matrix H is fundamentally used for understanding what each latent topic represents and to show the importance of each term for these latent topics. To interpret a topic, one typically analyzes a ranked list of the most probable terms in that topic. The problem with interpreting topics this way is that common terms in the corpus often appear near the top of such lists for multiple topics, making it hard to differentiate the meanings of the topics. Also, this approach alone doesn't provide us a way to make a direct connection between documents and terms.

Taddy [39] defined a quantity called *lift* to rank the terms within a topic, which is the ratio of a term's probability within a topic to its margin probability across the corpus. It gives higher rankings to rare terms that belong to specific topics and also decreases the rankings of globally common terms.

Bischof et al. [40] proposed a method to rank the terms for a given topic that uses the terms' frequency in that topic as well as the terms' *exclusivity* to the topic. A term's *exclusivity* to a topic is its frequency in that topic relative to a set of comparison topics. They developed hierarchical Poisson convolution model (a generative model

for labeled corpora that exploits the known topic hierarchy) to infer these relations and introduced a univariate scoring measure called *FREX* (Frequency-Exclusivity) to summarize performance in both frequency and exclusivity dimensions.

Likewise, Sievert et al. [41] proposed a measure similar to *exclusivity* that is called *relevance*. It is a weighted average of the logarithms of a term's probability under a topic and its *lift*. They also created a web-based interactive visualization tool called LDAvis to visualize topics estimated using LDA model and *relevance* measure.

Here we propose a similar measure called *purity* that benefits from the theme structure and the background topic of our *TSNMF* model to rank terms. As we have mentioned before, the terms with higher frequencies tend to have higher scores even though they do not represent the characteristics of a theme. It doesn't mean that these generic terms are not important, but one may want to observe terms that are more specific to a theme. We define the *purity* of term τ for subtopic i under theme k as:

$$purity\,(k, i, \tau) = \frac{\hat{H}^k_{i,\tau}}{\hat{H}^k_{i,\tau} + \hat{H}^k_{\neg I,\tau}} \tag{6.1}$$

where $\neg I$ represents the background topic. $\hat{H}$ matrix is the normalized version of H matrix over terms for each topic. $\hat{H}^k_{i,\tau}$ is the raw score of term τ for subtopic i under theme k that represents how important term τ is for subtopic i. $\hat{H}^k_{\neg I,\tau}$ is the score of term τ for the background topic of theme k that represents how important term τ is for all the themes in the dataset. It is important to remember that our main *Separated TSNMF* model generates a different theme-term matrix, $\hat{H}^k$, for each theme k. So, $\hat{H}^k_{\neg I,_}$ vector will be different for each theme, but will be the same for every subtopic in a theme. (For the *Combined TSNMF* model, there is only one topic-term matrix, $\hat{H}$, and because of that we have only one $\hat{H}^k_{\neg I,_}$ vector for all the topics.)

For simplicity and easier further use, let's call $\hat{H}^k_{i,\tau}$ as *Direct Term Score (DTS)* of term τ for subtopic i under theme k and $\hat{H}^k_{\neg I,\tau}$ as *Background Term Score (BTS)* of

term τ for theme k. We can also define *purity* using *DTS* and *BTS* as follows:

$$DTS\,(k,i,\tau) = \hat{H}^k_{i,\tau} \tag{6.2}$$

$$BTS\,(k,\tau) = \hat{H}^k_{\neg I,\tau} \tag{6.3}$$

$$purity\,(k,i,\tau) = \frac{DTS\,(k,i,\tau)}{DTS\,(k,i,\tau) + BTS\,(k,\tau)}. \tag{6.4}$$

Purity can have values between 0 and 1. If we have a very common term in our hand, it means that it will probably have both high *DTS* and *BTS* and its *purity* will be around 0.5. If we have a term that is specific to our interested subtopic, then it will probably have a high *DTS* but a low *BTS* for that subtopic and its *purity* will be near 1. Last, if we have a term that is irrelevant to our interested subtopic, then it will probably have a low *DTS* but a high *BTS* for that subtopic and its *purity* will be near 0.

purity brings us to the definition of *Purity Term Score (PTS)* which is the last step before *TTS*. We define *PTS* of term τ for subtopic i under theme k as:

$$PTS\,(k,i,\tau) = purity\,(k,i,\tau)\,\hat{H}^k_{i,\tau}. \tag{6.5}$$

PTS is simply multiplication of a term's *purity* and its *DTS*. We use *purity* like a regularizer for *DTS* to fix the frequency problem. Finally, we define the *TTS* of term τ for subtopic i under theme k given a weight parameter λ called *purity* ratio (where $0 \leq \lambda \leq 1$) as:

$$TTS\,(k,i,\tau \mid \lambda) = (1-\lambda)\,\hat{H}^k_{i,\tau} + \lambda\; purity\,(k,i,\tau)\,\hat{H}^k_{i,\tau}, \quad or \tag{6.6}$$

$$TTS\,(k,i,\tau \mid \lambda) = (1-\lambda)\,DTS\,(k,i,\tau) + \lambda\; PTS\,(k,i,\tau) \tag{6.7}$$

where λ determines the weight given to the *DTS* of term τ for topic i under theme k relative to its *PTS*. Setting $\lambda = 0$ results in the traditional ranking of terms in decreasing order of their *DTS*, and setting $\lambda = 1$ ranks terms entirely by their *PTS* which is our regularized version of *DTS*.

It is important to note that, when a theme has more than one subtopic, the method will generate subtopic count of different term distributions for the theme instead of one inclusive term distribution. So we can analyze each subtopic separately. But if one wants to obtain a single term distribution for a theme, different approaches can be utilized:

- **Max:** For each term, take the maximum score over the subtopics in a theme.

$$TTS_{max}\left(k, \tau \mid \lambda\right) = \max_{i} TTS\left(k, i, \tau \mid \lambda\right) \tag{6.8}$$

- **Sum:** For each term, take the summation of all scores over the subtopics in a theme.

$$TTS_{sum}\left(k, \tau \mid \lambda\right) = \sum_{i} TTS\left(k, i, \tau \mid \lambda\right) \tag{6.9}$$

6.1.1. Scoring Terms for Each Document

TTS offers great help to understand and interpret topics. But when it comes to documents, there is no direct way to find defining terms of a document. The most apparent method is to find the topic distributions of the documents first. Then according to these distributions and *TTS* of terms, one can figure out the defining terms of the documents. Even though the methodology seems straightforward, it is not easy to derive a comprehensive ranking among terms for a document for such reasons:

- A document generally doesn't include all the terms in a dictionary. But term distributions of topics are over all the terms. A document may not include the highest-ranking terms for the topic that it belongs to.
- Every topic has a different term distribution on the same dictionary. A term τ can rank high for one topic i_1, but low for another topic i_2. However, a document ν may belong to both of these topics, i_1 and i_2, and has the term τ. So, it is tricky to evaluate the importance of the term τ for the document ν.

We propose a very simple method to overcome these problems. In our method, we only redefine our building blocks DTS and BTS of term τ in document ν for subtopic i under theme k as follows:

$$DTS\left(k,i,\tau,\nu\right) = \mathbb{1}(X_{\nu,\tau} > 0)\ \hat{W}^{k}_{\nu,i}\hat{H}^{k}_{i,\tau} \tag{6.10}$$

$$BTS\left(k,\tau,\nu\right) \quad = \mathbb{1}(X_{\nu,\tau} > 0)\ \hat{W}^{k}_{\nu,\neg I}\hat{H}^{k}_{\neg I,\tau} \tag{6.11}$$

where X is our main data matrix, $\mathbb{1}(X_{\nu,\tau} > 0)$ is an indicator function and $\hat{W}$ is the normalized coefficient matrix over topics for each document. X matrix holds the document-term relationship which is generally tf or tf-idf values of the terms for the documents. The indicator function $\mathbb{1}(X_{\nu,\tau} > 0)$ evaluates to 1 when $X_{\nu,\tau} > 0$ and 0 otherwise. $X_{\nu,\tau}$ being greater than 0 means that document ν includes term τ. So, if term τ doesn't exist in document ν, then its DTS and BTS evaluates to 0 and term τ is eliminated from the term ranking of document ν (even if the term τ is a high ranked term in a topic that document ν belongs to). The multiplication of $\hat{H}$ with $\hat{W}$ allows the term scores to change according to the documents' topic distributions.

Definitions of *purity*, PTS and TTS also change according to the new definitions of DTS and BTS as follows:

$$purity\left(k,i,\tau,\nu\right) \quad = \frac{DTS\left(k,i,\tau,\nu\right)}{DTS\left(k,i,\tau,\nu\right) + BTS\left(k,\tau,\nu\right)} \tag{6.12}$$

$$PTS\left(k,i,\tau,\nu\right) \quad = purity\left(k,i,\tau,\nu\right) DTS\left(k,i,\tau,\nu\right) \tag{6.13}$$

$$TTS\left(k,i,\tau,\nu \mid \lambda\right) = (1-\lambda)\, DTS\left(k,i,\tau,\nu\right) + \lambda\ PTS\left(k,i,\tau,\nu\right). \tag{6.14}$$

Besides the addition of the new parameter ν which corresponds to documents, the motivation behind *purity*, PTS and TTS stays the same.

It is important to note that, with Equation 6.7 we have generated a new score for each subtopic-term pair which corresponds to a 2D matrix as a whole. But with Equation 6.14, it is now a new score for each subtopic-term-document triplet which

corresponds to a 3D tensor as a whole. Each document will have a separate ranking for each subtopic. These rankings will only consist of terms that take place in the corresponding documents. Most importantly, rankings from different subtopics for a document will be comparable. Also, term distributions for themes can be obtained by applying the methods shown in Equations 6.8 or 6.9. As a result, with document-specific *TTS*, we will be able to rank the terms and identify the most important ones for documents.

7. PREPARATION FOR EXPERIMENTS

We are approaching the chapter of tables and plots. But, before diving into the excitement of the results, we will show behind the scenes. We will start with how documents are processed before entering the model, and continue with the datasets. Then, we will explain our evaluation metric along with the genetic algorithm that is introduced to make the traditional NMF and LDA models comparable to the *TSNMF* model.

7.1. Text Preprocessing

Before applying any major NLP techniques to the documents, we tried to fix some wording and text emoticons using a manually crafted dictionary, such as *"isn't"* to *"is not"* and *":)"* to *"smile"*. Then, each document is tokenized while removing stopwords, punctuation, numbers, and any token with fewer than 3 characters long. Tokens are lemmatized using Wordnet lemmatizer of Natural Language Toolkit (NLTK) [42] to obtain the base form of tokens. Finally, we removed documents that have less than 25 words.

To create a document-term matrix (data matrix), an n-gram word model is used combining unigrams, bigrams, and trigrams to generate a dictionary. To prevent over-populating the dictionary, only the most frequent 10,000 terms are considered. We used both tf and tf-idf (with L2-normalization) separately to encode documents into the document-term matrix in our experiments.

7.2. Datasets

We used 3 document datasets in our experiments: Brown Corpus, Reuters Corpus, and Schwartz's Theory of Basic Human Values Dataset. Table 7.1 summarizes the basic statistics of the datasets after applying text preprocessing.

Table 7.1. Basic statistics of the datasets after applying text preprocessing.

Dataset	# docs	# topics
Brown	500	15
Reuters	8,159	90
Schwartz	433	10

7.2.1. Brown Corpus

The Brown Corpus contains 500 documents, and the documents have been categorized by 15 topics, such as adventure, editorial, fiction, and so on. Every document is labeled with only one topic. The Brown Corpus is included in NLTK text corpora.

7.2.2. Reuters Corpus

The Reuters Corpus contains 10,788 (8,159 after text preprocessing) news documents that have been classified into 90 topics. Unlike the Brown Corpus, categories in the Reuters Corpus overlap with each other, simply because a news story often covers multiple topics. Thus, a document in the Reuters Corpus can have multiple topics. The Reuters Corpus is also included in NLTK text corpora.

7.2.3. Schwartz's Theory of Basic Human Values Dataset

The Theory of Basic Human Values, developed by Shalom H. Schwartz, tries to measure universal values that are recognized throughout all major cultures [43]. Schwartz's theory identifies ten such motivationally distinct values and also categorizes them in five higher order groups:

- **Openness to change**: *Self-Direction* and *Stimulation*.
- **Self-enhancement**: *Achievement* and *Power*.

- **Hedonism**: *Hedonism* (considered to be shared among *Openness to change* and *Self-enhancement*).
- **Conservation**: *Security*, *Conformity*, and *Tradition*.
- **Self-transcendence**: *Benevolence* and *Universalism*.

To better describe these relationships graphically, the theory organizes ten values in a circular structure shown in Figure 7.1.

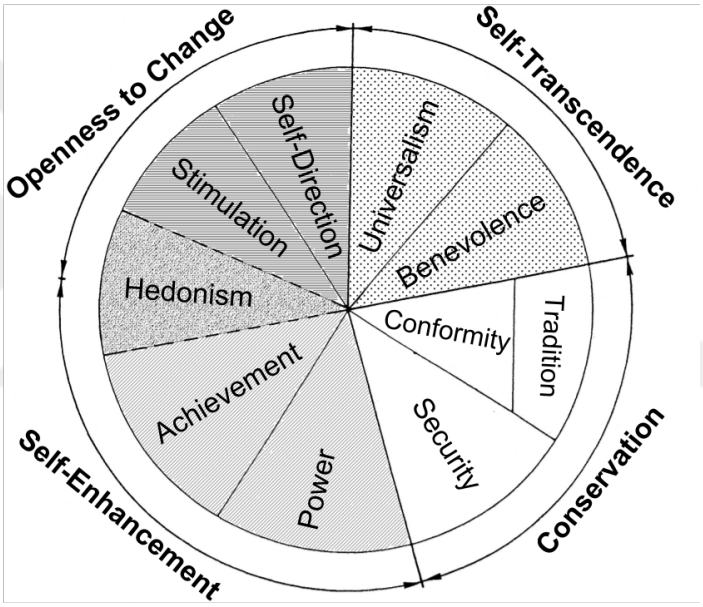


Figure 7.1. The relationships among the basic human values and higher order groups in Schwartz's Theory of Basic Human Values.

This dataset was created in one of our previous related study [44]. To obtain the dataset, Wikipedia articles were crawled for the ten basic human values. For each value, a few key seed articles were qualitatively selected to construct a value-specific corpus of Wikipedia articles. The Short Schwartz Values Survey which is a validated tool based on the original Schwartz Values Survey was used to select seed documents. A custom crawler was developed that exploits the structural characteristics of Wikipedia articles. It traverses the URLs within the *See also*, *Relevant topics* and *References* sections of documents. In this work, only the seed articles and articles that are one

hop away from the seed articles are used (433 documents). Table 7.2 shows the number of documents for each basic human value.

Table 7.2. Number of documents for each basic human value (BHV) in the Schwartz dataset.

BHV	# docs	**BHV**	# docs
Universalism	136	**Power**	21
Benevolence	45	**Achievement**	44
Conformity	32	**Hedonism**	76
Tradition	17	**Stimulation**	7
Security	28	**Self-direction**	27

7.3. Evaluation Metrics

Evaluating an unsupervised topic model is not a straightforward task. Because there is no certain correct output for an input. Without ground truth, there is always room for interpretation. One model can find more distinct topics, while another model can explain the topics better. We also couldn't find a standard evaluation metric for topic models in the literature.

In our problem setting, we have document labels in advance. So, we could use evaluation metrics that are used in supervised ML literature such as accuracy and F1 score. However, we don't think these metrics do justice for topic models. Because topics are not strictly separated objects and can be related to each other. For example, assume that we knew Document 1 is about Topic A, but the topic model gives Topic A the second-highest score for Document 1. In terms of accuracy, it is a misclassification. But if there are tens of topics, then having the second-highest score should not be regarded as a misclassification. Using the topic rankings of documents, we propose a novel evaluation metric called *log rank accuracy* to assess semi-supervised topic models.

It should be pointed out that the topic distribution of documents is just one side of the coin; on the other side, we have the term distribution of topics, also known as the topic descriptions. So, for a better assessment of topic models, one should also always analyze the topics themselves, and we will do it at the end of Chapter 8.

7.3.1. Log Rank Accuracy

Log rank accuracy evaluates a score for each document using the ranked list of themes according to their *theme scores* (Equation 5.1) to calculate the document's theme assignment accuracy. *Log rank accuracy* of each document is calculated as follows:

(i) Sort the themes in descending order using *theme scores*.

(ii) Find the ranks of the themes that are assigned to the document and store in the list R (ranking starts from 1).

(iii) Log rank score for the document is $\sum_{r \in R} \ln P - \ln r$, where P is the total number of themes of the dataset.

(iv) Calculate the maximum log rank score that the document can have and take the ratio of the log rank score of the document and the maximum possible log rank score to achieve the document's *log rank accuracy* as it is shown in Equation 7.1 (if a document has more than one theme, then we expect these themes to have the highest scores without any ordering between them).

$$\text{Log Rank Accuracy} \triangleq \frac{\sum_{r \in R} \ln P - \ln r}{\sum_{i=1}^{|R|} \ln P - \ln i} \tag{7.1}$$

The reason of the last step is that, for each document, the maximum possible score changes with respect to the number of themes in the dataset and the number of themes that the document assigned to: A document with one assigned theme can get the maximum score of $\ln P$ where P is the total number of themes. But no one likes scoring schemes that do not have a fixed scale, because they are harder to interpret and compare with each other. So we fixed the scoring range between 0 and 1 with the

last step where *log rank accuracy* of 1 means perfect theme assignment.

Here is an example to show how *log rank accuracy* works: Assume that we have 20 themes and 2 documents (Document 1 and Document 2) in our test dataset. We knew that Document 1 is labeled with only Theme A, while the labels of Document 2 are both Theme A and Theme B. When we run our model for Document 1 and Document 2, we get the following outputs:

- **Document 1:** The model says Theme A has the second highest score for Document 1. Log rank score of Document 1 becomes $\ln 20 - \ln 2$. Because Document 1 has only one theme, the max possible log rank score for it is $\ln 20 - \ln 1$. To calculate the *log rank accuracy*, we take the ratio of these two scores: $\frac{\ln 20 - \ln 2}{\ln 20} = 76.9\%$.
- **Document 2:** The model says Theme B has the highest score and Theme A has the third-highest score for Document 2. Log rank score of Document 2 becomes $(\ln 20 - \ln 1) + (\ln 20 - \ln 3)$. Because Document 2 has two themes, the max possible log rank score for it is $(\ln 20 - \ln 1) + (\ln 20 - \ln 2)$. To calculate *log rank accuracy*, we take the ratio of these two scores: $\frac{(\ln 20 - \ln 1) + (\ln 20 - \ln 3)}{(\ln 20 - \ln 1) + (\ln 20 - \ln 2)} = 92.3\%$.

We used a logarithmic function that utilizes the ranking of the themes to solve the problem that we have mentioned about the accuracy. Instead of a logarithmic function, one can also consider using a linear function. But, we wanted to reflect the idea that the importance of a rank should decrease when the rank gets lower (worse). In other words, the difference between being the first or second should be more important than the difference between being the last or second from the last. Contrary to a normal logarithm function, our *log rank accuracy* function is convex and acts like an exponential function; because we are subtracting logarithm of each rank from the logarithm of the number of themes. And this property enabled the change in the *log rank accuracy* in the higher ranks to be more than the change in the lower ranks (see Figure 7.2). For example, the difference between being rank 1 and rank 2 is greater than the difference between being rank 19 and rank 20 for a theme that should be in the first place. To calculate the *log rank accuracy* of a dataset, we take

the average of *log rank accuracy* of all the documents in it.

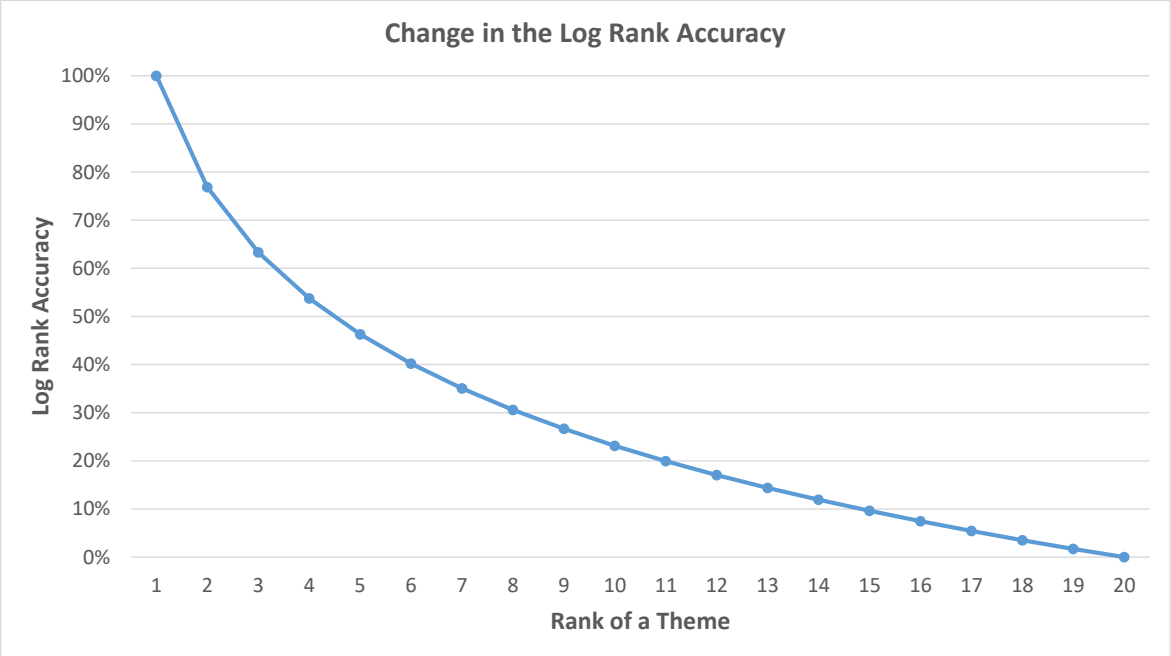


Figure 7.2. Change in the *log rank accuracy* with respect to the change in the ranking of an assigned theme in a dataset with 20 themes.

7.3.2. Genetic Algorithm for Traditional NMF and LDA

After applying traditional NMF or LDA to a corpus, it is not an easy task to identify what the topics correspond to. Under unsupervised conditions, a human expert needs to find a meaningful label for each topic. But in our problem setting, the topics are predefined. So, to be able to compare the *TSNMF* with the traditional NMF and LDA we need to find the best possible matching between the topics discovered by the traditional NMF or LDA and the predefined topics.

When the number of topics is too small, the matching problem is not a hard task to accomplish for a human. But the problem gets exponentially bigger with the number of topics because the number of possible matching is the factorial of the number of topics. To solve this combinatorial matching problem, we propose a metaheuristic called genetic algorithm [45] that is inspired by the process of natural selection where

the fittest individuals are selected for reproduction to produce offspring of the next generation.

The process of natural selection usually starts from a randomly generated population where each iteration of the population is called generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population as parents. The parents produce offspring which inherit the modified (recombined and mutated) characteristics of the parents. The offspring are then used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached. A typical genetic algorithm requires two objects:

- **Solution Representation:** We used an integer list for the permutation of topic ids as the representation. The list represents the topic matching. The elements and the indices of the list correspond to the topic ids of the traditional NMF or LDA and the predefined topic ids respectively. For example; the list $[3, 2, 0, 1]$ implies that Topic 3 of traditional NMF corresponds to the predefined Topic 0, Topic 2 of traditional NMF corresponds to the predefined Topic 1, and so on.
- **Objective and Fitness Function:** We used *log rank accuracy* as our objective and fitness function. The goal was to maximize the *log rank accuracy*.

Our genetic algorithm consists of 6 steps:

(i) **Initialization:** Generate M number of individuals for the initial population using a greedy approach. For each individual, visit their genes (indices of the solution) in random order and assign topic ids that give the highest fitness.

(ii) **Selection:** Select M pairs of individuals (parents) from the current population

according to the selection probability:

$$p(k) = \frac{2k}{M(M+1)} \tag{7.2}$$

where M is the number of individuals and k is the k^{th} individual in ascending order of fitness. This implies that the median value has a chance of $\frac{1}{M}$ of being selected, while the M^{th} (the fittest) has a chance of $\frac{2}{M+1}$, roughly twice that of the median.

(iii) **Crossover:** Apply two-point crossover to each of the selected pairs in Step 2 to generate M offspring with the crossover probability P_c. If the crossover operator is not applied, then one parent remains as the new offspring. In the two-point crossover, two points are randomly selected for dividing one parent. The numbers outside the selected two points are always inherited from one parent to the child, and the other numbers are placed in the order of their appearance in the other parent (see Figure 7.3).

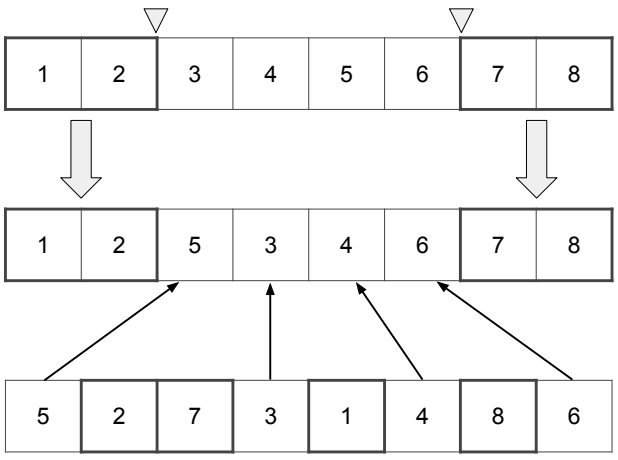


Figure 7.3. Two-point crossover

(iv) **Mutation:** Apply shift change mutation to each of the produced M offspring with the mutation probability P_m. If the mutation operator is not applied, the offspring remains the same. In shift change mutation, a number at one position is removed and put at another position. Then all other numbers shifted accordingly.

The two positions are randomly selected. (see Figure 7.4).

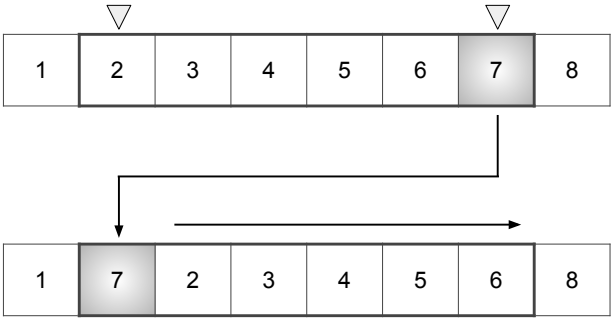


Figure 7.4. Shift change mutation

(v) **Elitist Update:** Randomly remove one individual from the current population and add the best individual from the previous population to the current one.

(vi) **Termination:** Total number of generations used as the stopping condition.

8. EXPERIMENTS AND RESULTS

We have finally reached the chapter of tables and plots. In this chapter, we first determined the best parameter settings for our models to not overpopulate the figures and tables. To increase the consistency of the results, we repeated each training process 5 times with different train-test sets similar to the k-fold cross-validation, and then we presented the results as the average of these 5 experiments. While splitting train-test sets, we always made sure that each topic was represented in both sets at least by one document. To measure the effect of supervision, we tested our models with different supervision ratios that are the ratio of the training set size to the whole corpus size. Except for the last section, we used 3 subtopics for the themes and *log rank accuracy* as the evaluation metric to compare the results of the models. After choosing the best parameters, we conducted 3 experiments that are about:

- The comparison of *bCool* initialization with random initialization.
- The comparison of *TSNMF* models among themselves and with traditional NMF and LDA.
- The effect of our proposed hierarchical topic structure and term scoring methods (*purity* measure) on the interpretation of topics.

We used both *Separated* and *Combined TSNMF* models in the experiments. The models are trained in both semi-supervised and supervised fashions which are explained in Sections 3.5 and 3.6 in detail. To remind briefly, in semi-supervised training, all the documents in the dataset enter the training process and there is no testing phase; because the model discovers the theme distributions of unlabeled documents in the training phase. On the other hand, in supervised training, only the labeled documents enter the training process and the theme assignments of the unlabeled documents are handled in a separate testing phase. The four *TSNMF* models that have been used in the experiments are as follows:

- Semi-supervised *Separated TSNMF* (demonstrated in Figure 3.5)
- Semi-supervised *Combined TSNMF* (demonstrated in Figure 3.4)
- Supervised *Separated TSNMF* (demonstrated in Figure 3.6)
- Supervised *Combined TSNMF*

8.1. Finding The Best Parameter Settings

We start the experiments by deciding which cost function (square of Euclidean distance or KL divergence) and text encoding technique (tf or tf-idf) to use for each model and method. We didn't include the combination of the square of Euclidean distance and tf-idf because it didn't produce meaningful results for our model. Square of Euclidean distance with the tf got the worst scores, while KL divergence with tf had the edge over KL divergence with tf-idf. Actually, KL divergence with tf-idf got higher scores than KL divergence with tf for some settings, but we decided to use tf encoding for two reasons: it is simpler than tf-idf and background topics in our model try to accomplish a similar idea with tf-idf.

Table 8.1 shows the comparison between each setup for *Separated TSNMF* model with *bCool* initialization on Brown corpus with the supervision ratios of 30% and 70%. We can see that the KL divergence with tf encoding always has the edge over the other two setups. It is important to mention that, we didn't decide to use KL divergence with tf by only analyzing this table. We conducted every possible configuration on every dataset and decided by analyzing all of them together. Table 8.1 is just a sample from all these results.

8.2. How to Initialize: bCool vs Random

We tested the impact of *bCool* initialization against random initialization. Figure 8.1 shows the results in three separate bar charts for three datasets. In the charts, heights of the bars represent the difference between *log rank accuracy* of *bCool* and random initialization; and supervision ratio increases from left to right.

Table 8.1. The comparison between 3 different cost function and text encoding combinations. Scores are obtained using *Separated TSNMF* model with *bCool* initialization on Brown corpus.

Method	**Supervision**	**Cost**	**Encoding**	**Log Rank Accuracy**
Semi-supervised	30%	Euclidean	tf	59.26%
Semi-supervised	30%	KL	tf	**64.06%**
Semi-supervised	30%	KL	tf-idf	62.06%
Supervised	30%	Euclidean	tf	65.23%
Supervised	30%	KL	tf	**69.78%**
Supervised	30%	KL	tf-idf	69.76%
Semi-supervised	70%	Euclidean	tf	68.43%
Semi-supervised	70%	KL	tf	**74.05%**
Semi-supervised	70%	KL	tf-idf	72.73%
Supervised	70%	Euclidean	tf	73.23%
Supervised	70%	KL	tf	**75.67%**
Supervised	70%	KL	tf-idf	74.33%

From charts, we can see that semi-supervision heavily favors *bCool* initialization over random initialization; especially for the semi-supervised *Separated TSNMF*. This extreme difference is due to the random initialization working very poorly for the model, not because *bCool* for this model gives better results than any other model. For the supervised setup, random initialization seems to perform slightly better; especially at low supervision ratios.

The reason for *bCool* to perform better than random initialization with semi-supervision and worse in lower supervision ratios with supervised models stems from the same particular operation in *bCool*: When the supervision ratio is low, the number of documents per theme decreases and this causes the partitions to not have enough documents. To solve this problem, we assign documents randomly from any theme

with direct proportion to their densities. For this last part, for the supervised setup, the documents to be selected are restricted with the training (labeled) documents; for the semi-supervised setup, because all the documents are in the model, the documents can be selected from the whole corpus. While this is a great advantage for the semi-supervised setup, it creates a small disadvantage for the supervised setup at the low supervision ratios. So, one can use random initialization for supervised models in lower supervision ratios and switch back to *bCool* initialization in higher supervision ratios for other benefits of *bCool*.

Besides scoring, when we use *bCool*, the models converge at least 2 times faster than the random initialization. For the following Experiment sections, we will always show the results that we obtained using *bCool* initialization.

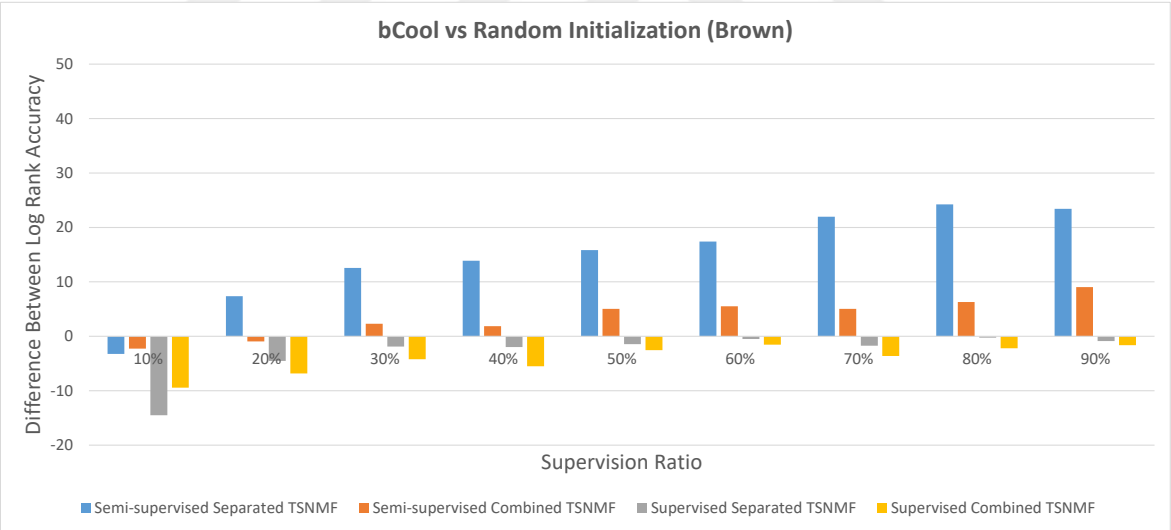
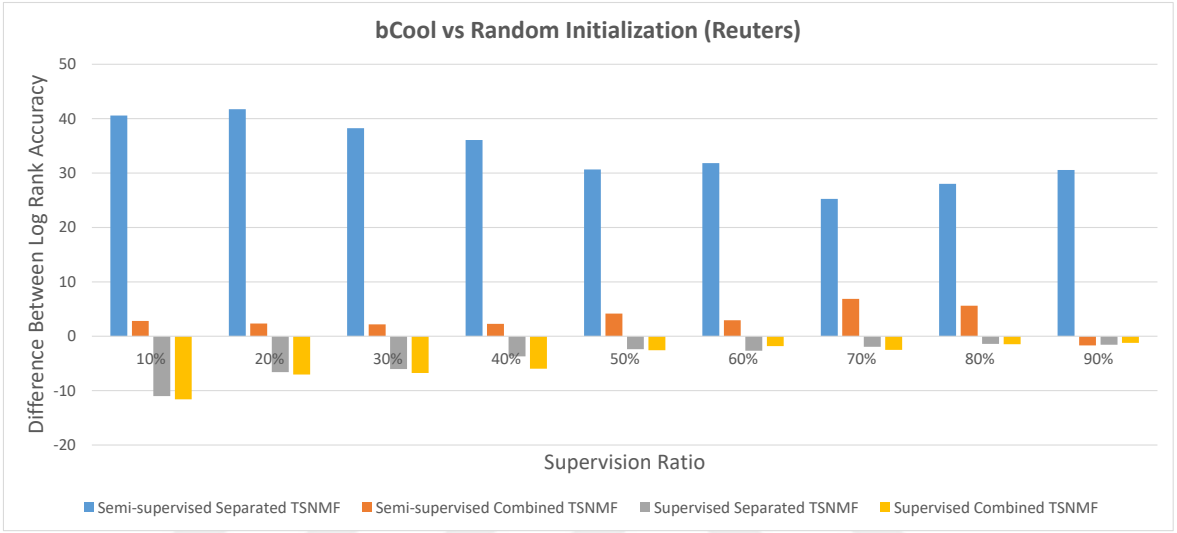


Figure 8.1. Difference between *log rank accuracy* of *bCool* and random initialized models on all datesets.



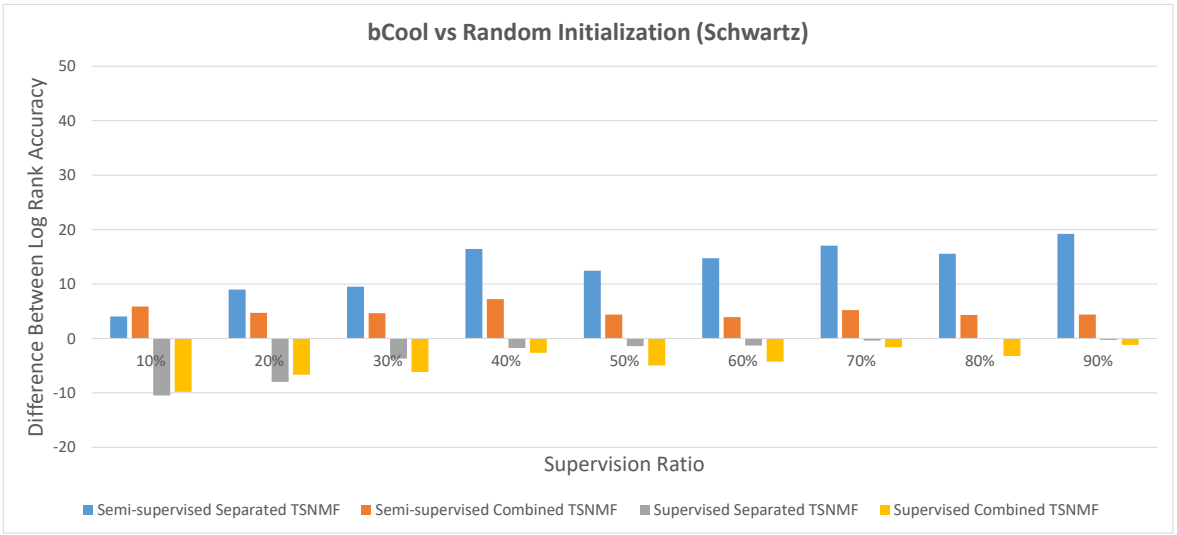


Figure 8.1. Difference between *log rank accuracy* of *bCool* and random initialized models on all datesets. (cont.)

8.3. How to Model: Novel vs Traditional

We ran all four of our *TSNMF* models on all three datasets with *bCool* initialization, KL divergence cost function, tf encoding, and 3 subtopics. We tested the models for different supervision ratios. To generate baseline scores, we also ran traditional NMF and LDA on the same setups. Because NMF and LDA are unsupervised methods, the training (decomposition) part was the same for all the supervision ratios. Without any labeling or theme hierarchy, all the documents are fed into the model and trained. The slight difference in scores between the supervision ratios came from the genetic algorithm that we used to find topic matching. Our genetic algorithm tries to find the optimal match between the topics found by the traditional NMF or LDA and the document labels that maximize the *log rank accuracy*. In the genetic algorithm, to find the best topic matching, we only used the training data that is restricted by the supervision ratio. Using supervision in the genetic algorithm didn't affect the final results that much, because having one correctly classified document is enough for the algorithm to make the right matching. Increasing the ratio just helped the algorithm to be more robust. But we wanted to make everything as even as possible.

Table 8.2 shows the *log rank accuracy* of every experiment for different supervision ratios for Brown, Reuters, and Schwartz datasets, respectively. Figure 8.2 demonstrates the results as line plots. We got very promising results and here are the main takeaways:

- Effect of supervision is noteworthy, especially at low ratios. But the effect decreases with the increasing ratio. Even if we have a small set of labeled documents, it makes a huge improvement. Particularly for large datasets such as Reuters where 20% supervision ratio helped us to identify nearly the whole dataset. So, the improvement may not be parallel to the ratio of training set size and dataset size but the number of documents in the training set.
- As the supervision ratio increases, the *log rank accuracy* does not always increase, occasional slight decreases may occur. The main reason for this is that the test set shrinks and misclassification becomes much more expensive. We see that the most

Table 8.2. *Log rank accuracy* of *TSNMF* models and the traditional NMF-LDA models on Brown, Reuters, and Schwartz datasets, respectively. The top row indicates the change in supervision ratio. GA corresponds to our genetic algorithm. Scores are in percentage format.

Brown	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Ss S-TSNMF	38	55	64	67	71	72	74	76	73
Ss C-TSNMF	**54**	63	67	66	70	71	71	72	72
S S-TSNMF	48	**64**	**70**	**71**	**73**	**74**	**76**	**78**	**76**
S C-TSNMF	53	62	67	68	70	72	73	74	74
NMF with GA	46	49	48	48	49	51	51	50	48
LDA with GA	41	45	45	48	46	49	51	50	47

Reuters	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Ss S-TSNMF	**79**	**86**	**87**	**89**	**90**	**91**	**92**	**90**	**89**
Ss C-TSNMF	43	44	45	47	50	50	56	57	54
S S-TSNMF	74	81	83	86	**89**	**90**	**91**	**90**	**89**
S C-TSNMF	64	69	69	72	74	76	77	76	74
NMF with GA	31	30	31	35	35	34	38	30	34
LDA with GA	37	36	35	35	33	34	37	38	36

Schwartz	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Ss S-TSNMF	53	66	70	**79**	**81**	81	84	84	85
Ss C-TSNMF	60	66	69	74	75	75	77	78	78
S S-TSNMF	**64**	**70**	**76**	**80**	**82**	**83**	**86**	**87**	**87**
S C-TSNMF	60	67	70	75	76	77	79	80	82
NMF with GA	53	53	53	54	54	56	55	52	55
LDA with GA	52	54	54	57	56	53	52	56	58

Ss: Semi-supervised, S: Supervised, S-TSNMF: Separated TSNMF, C-TSNMF: Combined TSNMF

successful results are around 70% supervision ratio and this is expected because while training any ML algorithm, it is a best practice to divide the dataset into 70% train and 30% test sets. Another possible reason is that with the changing supervision ratio, train and test documents also change; and worse documents (which have less information) may have come to the training set, which may have reduced the score. However, these fluctuations seem to be negligibly low.

- *Separated* models give much better results than *Combined* models. This was something we expected because the separation was already a step we took to develop the model and it is reflected in these results. *Separated* models are better than *Combined* models since they can use the background topic and the document scoring (see Chapter 5) method more effectively.
- Supervised models give much better results than semi-supervised models. While supervised models have some disadvantages such as not being able to learn new topics and not being able to access the terms of the documents in the test set; the results show that the supervised models can learn themes much better and match them correctly for documents. On the other hand, we can see the shortcomings of the supervised model from getting lower scores than the semi-supervised models sometimes in places where the supervision ratio is low. But still, if we are not aiming to discover a new topic, the supervised models give very accurate results.
- Genetic algorithm matches the topics of traditional methods with the predefined labels near optimal. We could understand it by comparing the *log rank accuracy* of traditional methods and semi-supervised methods at lower supervision ratios. Because semi-supervised version is more similar to the traditional methods than supervised version. Except Reuters Corpus (for the reason we explained in the first item), traditional methods got similar results. We also checked the topic matchings manually and we have only managed to make little improvements (2% - 3%).

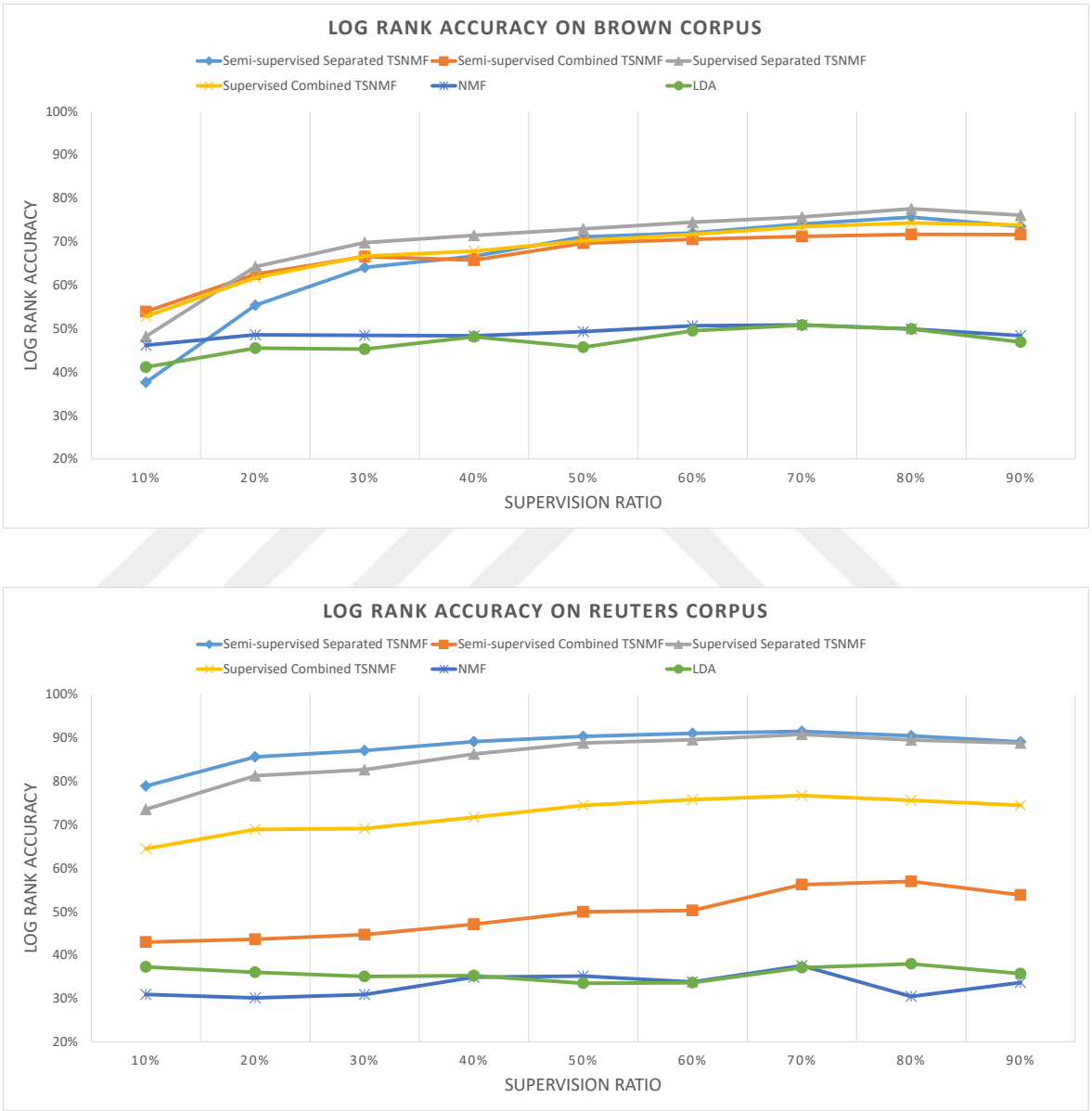


Figure 8.2. Plots of *log rank accuracy* of *TSNMF* models and traditional NMF-LDA on all datasets with respect to the change in supervision ratio.

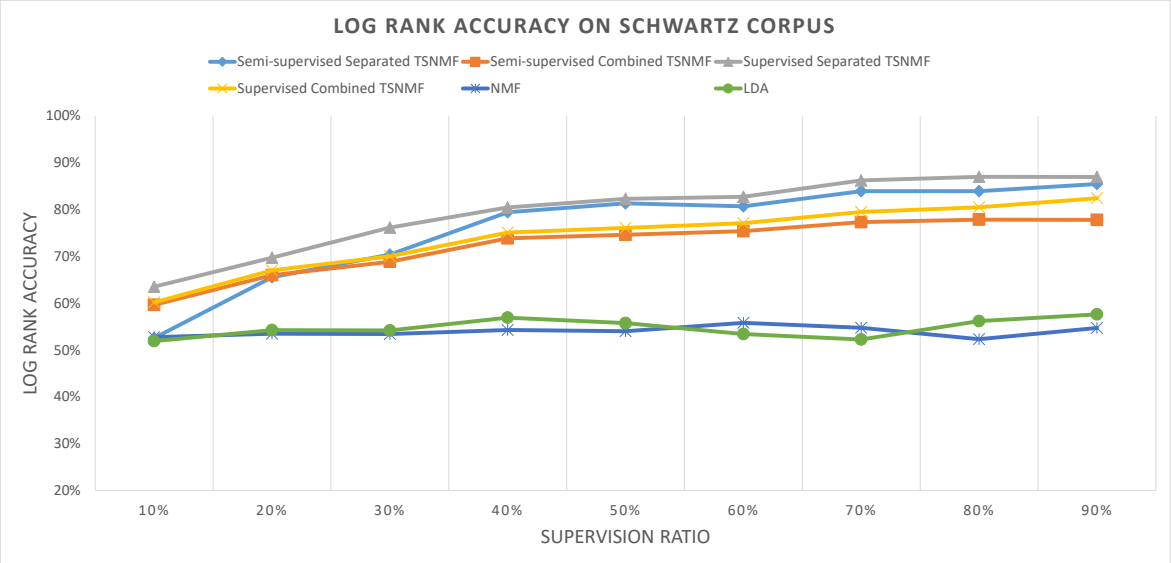


Figure 8.2. Plots of *log rank accuracy* of *TSNMF* models and traditional NMF-LDA on all datasets with respect to the change in supervision ratio. (cont.)

8.4. How to Interpret: Themes vs Topics

In previous sections, we compared different models in terms of classification performance using *log rank accuracy*. In this section, we will change our focus to the interpretation of the themes. In topic modeling, topics are represented with term distributions. In traditional NMF, because of its unsupervised nature, one needs to analyze each topic's term distribution to understand what they correspond to. For our *TSNMF* model, we determine this relation before the training starts. So without the need to make guesses, we knew which topic corresponds to which term distribution after the training. For any topic model, unsupervised or supervised, the information that we obtain from the topic-term relation is vital. Because although the main objective of topic models seems to be the categorization of documents, what gives them the edge is that topic models can also express this relationship between topics and terms. We used 5 different model setups to show how our proposed theme-subtopic structure and term scoring methods (*purity* measure) improve the term distributions of the themes:

- Traditional NMF.
- *Separated TSNMF* with 1 subtopic and *purity* ratio of 0.
- *Separated TSNMF* with 1 subtopic and *purity* ratio of 1.
- *Separated TSNMF* with 3 subtopics and *purity* ratio of 0.
- *Separated TSNMF* with 3 subtopics and *purity* ratio of 1.

For demonstration, we used universalism and hedonism themes from Schwartz dataset. Universalism is the understanding, appreciation, tolerance, and protection for the welfare of all people and for nature. Hedonism is pleasure or sensuous gratification for oneself. It is important to know these definitions to be able to interpret the results. We chose universalism and hedonism because they have much more distinct meanings and also have more documents in the dataset than other Schwartz values. All *TSNMF* models were trained with 100% supervision to generate the most accurate term distributions. We calculated *TTS* of each term for each topic-term distribution, and then the topmost 5 terms were chosen to represent the corresponding theme or subtopic. In the figures below, blue bars next to the terms represent the relative importance of the term for the model. A longer bar corresponds to a term that has a higher *TTS* for that model. Every model was scored separately from each other. *Purity* measure and *TTS* are the term scoring methods that we have introduced in Chapter 6 and the formulation can be found in Equation 6.7.

Figure 8.3 shows 4 different topics' topmost 5 terms for the traditional NMF. Because the traditional NMF is unsupervised, we have to match the term distributions with our themes. However, there was no clear term distribution representing any of our themes. Instead, we found two candidate topics for each of both the universalism and hedonism themes. Other than some common terms such as state, one, and human; and unrelated terms to hedonism such as law and court, the terms are pretty determinative. However, if those 4 distributions only belong to 2 themes, then it means that we need to match 6 distributions with the remaining 8 themes for Schwartz dataset. This situation reveals the problem of traditional NMF. Topics are blended in term distributions and this makes it very difficult to identify them.

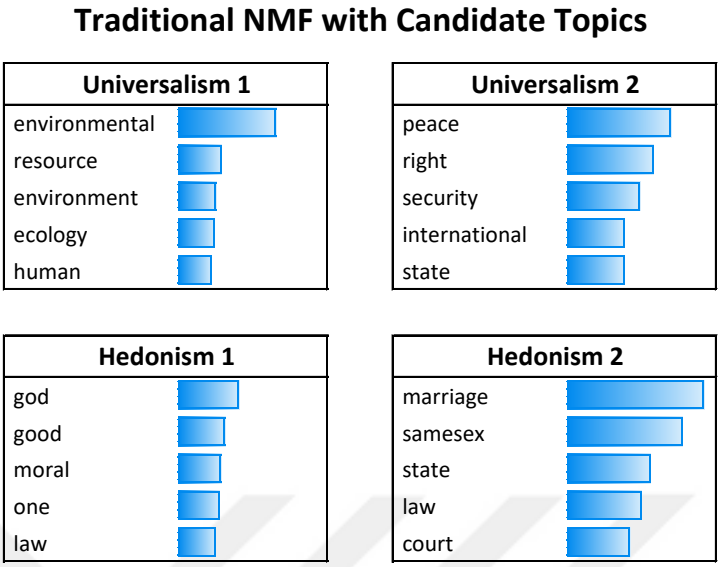


Figure 8.3. The topmost 5 terms of four topics from Schwartz dataset that could be interpreted as universalism and hedonism themes for the traditional NMF model.

In the figures of *TSNMF* models below, terms of background topics are not represented. Because, for all the models and themes, background topics gave more or less the same term distribution and we didn't want to populate the figures unnecessarily by giving background topics' term distributions for every figure. So, before moving on to the results of *TSNMF* models, here are the 9 topmost terms of background topics:

- one
- social
- use
- also
- may
- state
- human
- people
- individual

Figure 8.4 shows topmost 5 terms for the *Separated TSNMF* models with 1 subtopic and *purity* ratio of 0 and 1, respectively. Thanks to supervision, we know exactly which term distributions correspond to universalism and hedonism themes. Since the models have only 1 subtopic, there is one term distribution for universalism and one term distribution for hedonism.

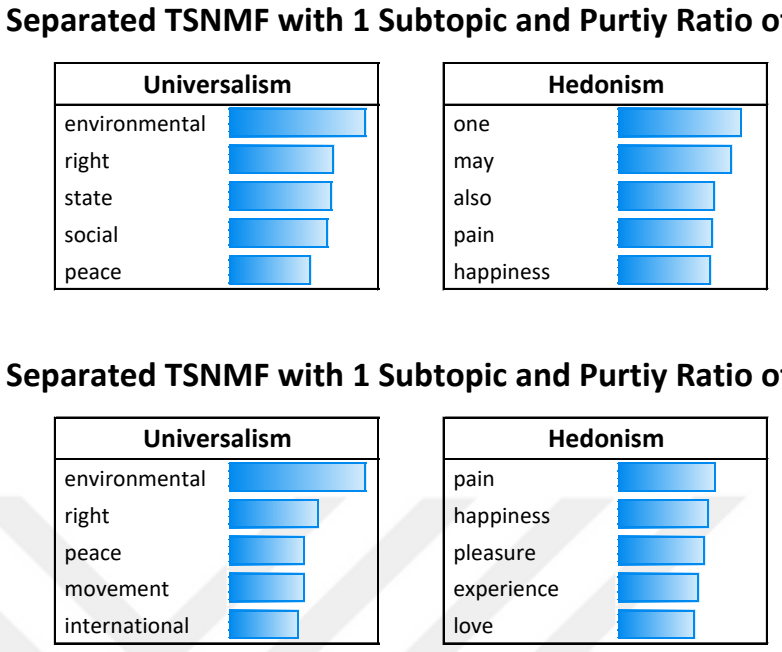


Figure 8.4. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* models with 1 subtopic and *purity* ratio of 0 and 1.

In Figure 8.4, when the *purity* ratio is 0, the topmost 3 terms of the hedonism theme belong to the background topic. Universalism does a better job by having a good highest scored topmost term, but it also has 2 terms that belong to the background topic. When we increase the *purity* ratio to 1, all the terms that belong to the background topic vanish while maintaining the orders of the significant terms. Our term scoring method cancels the noise caused by the frequent but insignificant terms and present tailor-made term distributions for the themes. This example demonstrates that our term scoring method is working as intended by improving the interpretation of the themes via generating better term distributions.

Figures 8.5 and 8.6 show topmost 5 terms for the *Separated TSNMF* models with 3 subtopics and *purity* ratio of 0 and 1, respectively. Having 3 subtopics enabled these models to represent each theme with 3 different term distributions. In addition to these, we obtained 2 more term distributions by aggregating subtopic-term distributions with maximum and summation operations to represent a theme with one term distribution using Equations 6.8 and 6.9, respectively.



Figure 8.5. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* model with 3 subtopics and *purity* ratio of 0.

The theme-subtopic structure reflects different aspects of themes with each subtopic. This distinction enables us to interpret themes much better and classify documents in more detail. The first subtopic of the universalism theme is about collective struggles such as war and peace, while the second subtopic is about equality and the third subtopic is about the environment. For the hedonism theme, we can say that the first subtopic is about personal feelings and pleasures, and the second subtopic is more about social emotions. However, it is hard to determine a meaningful label for the third subtopic, because the scores of all the terms are pretty low and close to each other, and also the topmost terms are all from the background topic.

The effect of *purity* ratio can again be clearly observed from the term distributions of the second and third subtopics of the hedonism theme as in the previous demonstration. The effect is more apparent for the sum aggregation where the terms from the background topic dominate the rankings for the model with *purity* ratio of 0. The reason for that is, even though background topic terms are not the dominant terms for the subtopics, they have consistently high scores for all the subtopics which cause their summation to pass the scores of the terms that high scores in one subtopic. So, the sum aggregation gives higher ranks to the terms that have relatively high scores for all the subtopics. While it fails miserably for the model with *purity* ratio of 0 as we have mentioned; it works very well for *purity* ratio of 1 where we can observe terms from different subtopics in high rankings. On the other hand, max aggregation picks each term's highest score among the subtopics. This method generates successful term distributions even for the model with *purity* ratio of 0. Because, our model still gives the top scores to theme-specific terms, even though term rankings are mostly filled with background topic terms. For the *purity* ratio of 1, if the score gap between high ranked terms of different subtopics is considerable (like hedonism), then the ranking prone to the most represented subtopic; but if the subtopics are more or less evenly distributed in terms of score, then the max aggregation gives similar term rankings as summation (like universalism).

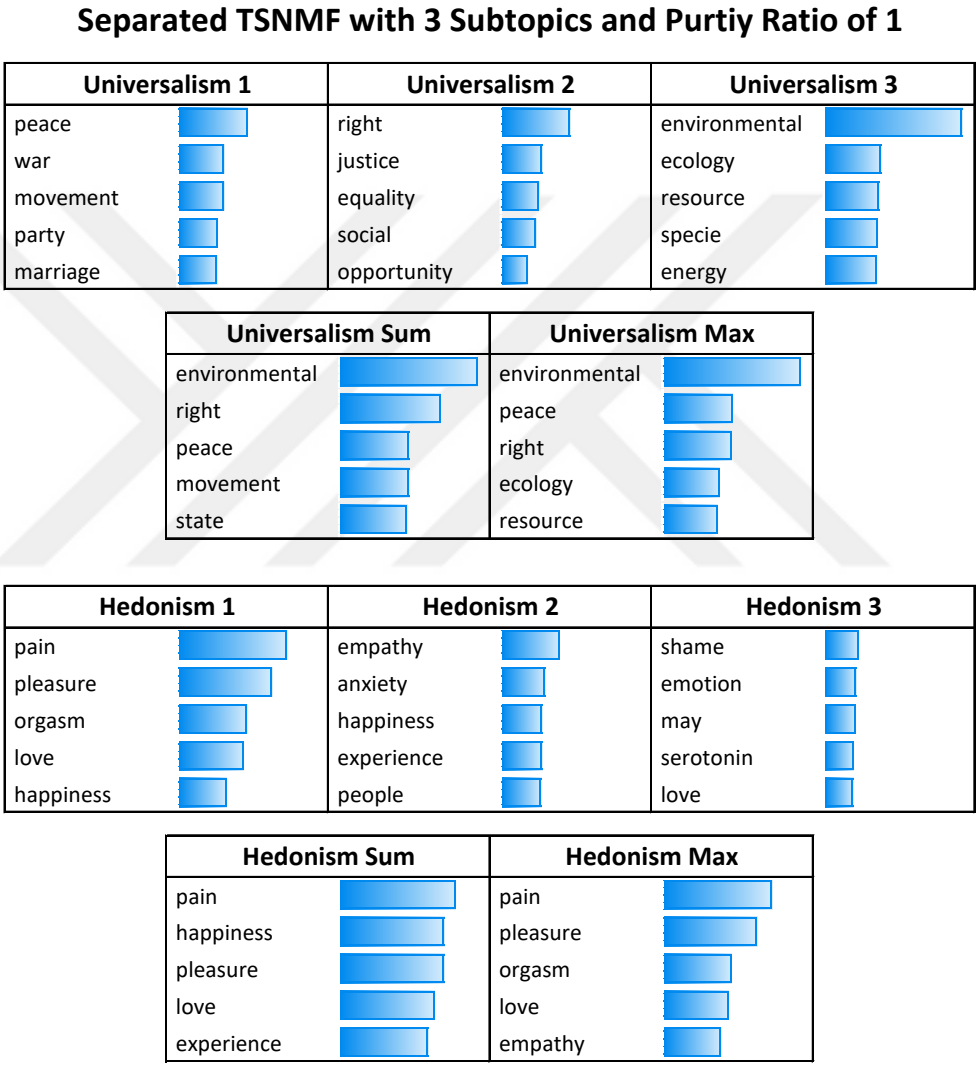


Figure 8.6. The topmost 5 terms of universalism and hedonism themes from Schwartz dataset for the *Separated TSNMF* model with 3 subtopics and *purity* ratio of 0.

9. CONCLUSION AND DISCUSSION

In this thesis, we proposed a semi-supervised topic model based on NMF called *Theme Supervised Nonnegative Matrix Factorization (TSNMF)* that can utilize labeled documents to improve the interpretation of the topics. *TSNMF* constrains the representation of the topics to align with the labeled documents and enables the topics discovered by the model to be readily understood. This approach eliminates the dependence on human interpretation on the outputs of traditional unsupervised topic models. We made it is possible to train *TSNMF* fully supervised with only labeled documents or semi-supervised with the labeled and unlabeled documents together. In fully supervision, using only labeled documents results in better-identified topics but the model can not discover new topics. On the other hand, semi-supervision allows the discovery of new topics and have an expanded term dictionary. The part so far was the semi-supervised topic model definition that could be found as a standard in the literature. Then, we introduced different structures and methods to improve this standard model.

Our novel hierarchical topic structure that consists of themes and subtopics is the key element of *TSNMF* as its name suggests. Creating unsupervised subtopic layers under the themes enabled the model to discover different aspects of themes. Then we combined the hierarchical topic structure with the separated training process and the background topics, for our model to reach its full potential. The background topic tries to generalize all the themes and acts as a regularizer in our scoring metrics. In separated training process every theme is trained separately similar to one-vs-all approach via utilizing background topics (theme-vs-background).

To generate more consistent and less varying factors, we proposed a new semi-deterministic initialization method called *bCool* that exploits the structure of *TSNMF*. Besides consistency, *bCool* sped up the training process at least 2 times and improved the results for the semi-supervised versions of the model over random initialization. For

the fully supervised versions of the model, *bCool* gave worse results than the random initialization when the supervision ratio was low. The reasons are that when the supervision ratio is low for the fully supervised version, we end up with very few training documents to use for the initialization and we introduce some forced randomness. For future work, we could try to find a better algorithm for *bCool* to apply in this situation.

To evaluate the models, we introduced *log rank accuracy* that uses the logarithm function to calculate a score for each document according to the ranking of their themes. For the *log rank accuracy*, the penalty for misclassification decreases as the rank goes down. So the biggest drop in score occurs when we choose a theme that should be in the first place, in second place. The difference between choosing the same theme last or the second last is negligible. This approach has its advantages and drawbacks over using a linear scoring function. As a drawback, a linear function is simpler and it results in higher scores than our method. Because a theme that should be first is unlikely to be placed in a very low rank by *TSNMF* and the penalty will always be higher for *log rank accuracy* than a linear function. Although we knew we would get lower scores and it may not seem good on the thesis, we developed and used *log rank accuracy* because we think it better reflects the importance placed on ranking in real life.

It is natural to compare *TSNMF* with its ancestor methods NMF and LDA. But these methods are unsupervised. So, it was not that fair and possible to compare them with our semi-supervised approach. Because, unsupervised topic models need manual topic naming which corresponds to matching term distributions to document labels in our case. However, it was not feasible to do this matching for document sets with a large number of topics. Therefore, we wrote a genetic algorithm to match the resulting topics of traditional NMF and LDA with our document labels. The method was like a post-supervision process where we add supervision to the results instead of the training process.

For the experiments, we considered 4 different versions of *TSNMF* to demonstrate the effect of every step we took to build the model. To not overpopulate the thesis with plots and tables, we first decided the best parameter settings for these models and showed the results only using them. We chose KL divergence as the cost function and tf as the text encoding technique. Actually, tf-idf gave better results for some setups, but we decided to use tf encoding because of its simplicity, and also background topics in our model try to accomplish a similar idea with tf-idf.

For the experiments, we used Brown Corpus, Reuters Corpus, and Schwartz dataset. Brown Corpus has well-defined documents and labels, but it is relatively small. Reuters Corpus is a multi-labeled dataset and has more documents and labels but not as well-defined as Brown Corpus. Schwartz dataset is a product of our previous work, that has documents about Schwartz's Theory of Basic Human Values and was collected semi-automatically. We plan to test our model on many more different datasets.

We tried different supervision ratios in our experiments to find a balance between the labeling cost and improvement in the model. The effect of supervision was noteworthy, especially at low ratios. 30% to 40% ratio could be considered as a general balance point. After these ratios, the rate of improvement slows down. And at 70% ratio, the improvement reaches its limit. But for Reuters Corpus which is larger than the other two datasets, improvement limit was reached at much earlier supervision ratios. So, the improvement may not be parallel to the training set and dataset size ratio but the number of documents in the training set. In other words, the effect of supervision can be independent of the dataset size which makes sense; because similar to supervised ML, we are actually training a model. After the model learns from sufficient amount of labeled documents, it can be applied to datasets with any size. The crucial part is to have labeled documents that come from a balanced topic distribution in terms of quality and quantity. Another result is that the *separated* models are superior to the *combined* models; because *separated* models can benefit from the background topic more effectively. Also, supervised models gave much better results than semi-supervised models; since they learned the topics better without the interference

of the unlabeled documents.

Along with the new hierarchical topic structure, to improve the interpretation of the topics, we introduced a novel measure called *purity* and implemented a new scoring scheme called *Theme Term Score (TTS)* using *purity* that alters the weights of terms for each topic's term distribution. We analyzed universalism and hedonism themes from Schwartz dataset using different models. We observed that supervision in *TSNMF* solved the topic and term distribution matching problem that we were facing in traditional methods. Subtopics successfully discovered different aspects of themes with their unsupervised nature which opens up different kinds of analysis on both labeled and unlabeled documents such as more detailed labeling. However, the real improvement in interpretation happened when we applied our new term scoring method to the outputs. The weights of the terms that are significant to subtopics were increased, while the weights of the common terms which are related to the background topics were decreased. So, we ended up with better-identified topics. We also used aggregation methods such as summation and maximum to represent themes as one topic instead of several subtopics and especially the max aggregation summarized the themes over subtopics pretty good.

REFERENCES

1. Lloyd, S., "Least squares quantization in PCM", *IEEE Transactions on Information Theory*, Vol. 28, No. 2, pp. 129–137, 1982.

2. Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman, "Indexing by latent semantic analysis", *Journal of the American Society for Information Science*, Vol. 41, No. 6, pp. 391–407, 1990.

3. Hofmann, T., "Probabilistic latent semantic indexing", *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, 1999.

4. Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.

5. Lee, D. D. and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", *Nature*, Vol. 401, No. 6755, pp. 788–791, 1999.

6. Brunet, J.-P., P. Tamayo, T. R. Golub and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization", *Proceedings of the National Academy of Sciences*, Vol. 101, No. 12, pp. 4164–4169, 2004.

7. Cemgil, A. T., "Bayesian inference for nonnegative matrix factorisation models", *Computational Intelligence and Neuroscience*, Vol. 2009, 17 pages, 2009.

8. Guan, N., D. Tao, Z. Luo and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, No. 7, pp. 1087–1099, 2012.

9. Zhang, S., W. Wang, J. Ford and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization", *Proceedings of the 2006 SIAM Interna-*

tional Conference on Data Mining, pp. 549–553, SIAM, 2006.

10. Blei, D. M., "Probabilistic topic models", *Communications of the ACM*, Vol. 55, No. 4, pp. 77–84, 2012.

11. Chang, J., S. Gerrish, C. Wang, J. L. Boyd-Graber and D. M. Blei, "Reading tea leaves: How humans interpret topic models", *Advances in Neural Information Processing Systems*, pp. 288–296, 2009.

12. Zhu, X. J., *Semi-supervised Learning Literature Survey*, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2005.

13. MacMillan, K. and J. D. Wilson, "Topic supervised non-negative matrix factorization", *arXiv preprint arXiv:1706.05084*, 2017.

14. Lee, H., J. Yoo and S. Choi, "Semi-supervised nonnegative matrix factorization", *IEEE Signal Processing Letters*, Vol. 17, No. 1, pp. 4–7, 2009.

15. Wang, D., X. Gao and X. Wang, "Semi-supervised nonnegative matrix factorization via constraint propagation", *IEEE Transactions on Cybernetics*, Vol. 46, No. 1, pp. 233–244, 2015.

16. Liu, Y., R. Jin and L. Yang, "Semi-supervised multi-label learning by constrained non-negative matrix factorization", *American Association for Artificial Intelligence*, Vol. 6, pp. 421–426, 2006.

17. Chen, Y., M. Rege, M. Dong and J. Hua, "Non-negative matrix factorization for semi-supervised data clustering", *Knowledge and Information Systems*, Vol. 17, No. 3, pp. 355–379, 2008.

18. Suyunu, B., *GitHub Repository of Theme Supervised Nonnegative Matrix Factorization*, 2020, `https://github.com/suyunu/theme-supervised-nmf`, accessed in August 2020.

19. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

20. Lee, D. D. and H. S. Seung, "Algorithms for non-negative matrix factorization", *Advances in Neural Information Processing Systems*, pp. 556–562, 2001.

21. Burred, J. J., *Detailed Derivation of Multiplicative Update Rules for NMF*, 2014, `https://jjburred.com/research/pdf/jjburred_nmf_updates.pdf`, accessed in April 2020.

22. Rubenstein, H. and J. B. Goodenough, "Contextual correlates of synonymy", *Communications of the ACM*, Vol. 8, No. 10, pp. 627–633, 1965.

23. Dumais, S. T., "Latent semantic analysis", *Annual Review of Information Science and Technology*, Vol. 38, No. 1, pp. 188–230, 2004.

24. Griffiths, T. L. and M. Steyvers, "Finding scientific topics", *Proceedings of the National Academy of Sciences*, Vol. 101, No. suppl 1, pp. 5228–5235, 2004.

25. Hoffman, M., F. R. Bach and D. M. Blei, "Online learning for latent Dirichlet allocation", *Advances in Neural Information Processing Systems*, pp. 856–864, 2010.

26. Blei, D. M., M. I. Jordan *et al.*, "Variational inference for Dirichlet process mixtures", *Bayesian Analysis*, Vol. 1, No. 1, pp. 121–143, 2006.

27. Ding, C., T. Li and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing", *Computational Statistics & Data Analysis*, Vol. 52, No. 8, pp. 3913–3927, 2008.

28. Gaussier, E. and C. Goutte, "Relation between PLSA and NMF and implications", *Proceedings of the 28th Annual International ACM SIGIR Conference on Research*

and Development in Information retrieval, pp. 601–602, 2005.

29. Girolami, M. and A. Kabán, "On an equivalence between PLSI and LDA", *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 433–434, 2003.

30. Chen, Y., L. Wang and M. Dong, "Non-negative matrix factorization for semisupervised heterogeneous data coclustering", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1459–1474, 2009.

31. Fei, W., L. Tao and Z. Changshui, "Semi-supervised clustering via matrix factorization", *Proceedings of 2008 SIAM International Conference on Data Mining (SDM 2008)*, 2008.

32. Griffiths, T. L., M. I. Jordan, J. B. Tenenbaum and D. M. Blei, "Hierarchical topic models and the nested chinese restaurant process", *Advances in Neural Information Processing Systems*, pp. 17–24, 2004.

33. Petinot, Y., K. McKeown and K. Thadani, "A hierarchical model of web summaries", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 670–675, Association for Computational Linguistics, 2011.

34. Mao, X.-L., Z.-Y. Ming, T.-S. Chua, S. Li, H. Yan and X. Li, "SSHLDA: a semi-supervised hierarchical topic model", *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 800–809, Association for Computational Linguistics, 2012.

35. Mcauliffe, J. D. and D. M. Blei, "Supervised topic models", *Advances in Neural Information Processing Systems*, pp. 121–128, 2008.

36. Ramage, D., D. Hall, R. Nallapati and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora", *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 248–256,

2009.

37. Ding, C., X. He and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering", *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 606–610, SIAM, 2005.

38. Langville, A. N., C. D. Meyer, R. Albright, J. Cox and D. Duling, "Initializations for the nonnegative matrix factorization", *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 23–26, 2006.

39. Taddy, M., "On estimation and selection for topic models", *Artificial Intelligence and Statistics*, pp. 1184–1193, 2012.

40. Airoldi, E. M. and J. M. Bischof, "A Poisson convolution model for characterizing topical content with word frequency and exclusivity", *arXiv preprint arXiv:1206.4631*, 2012.

41. Sievert, C. and K. Shirley, "LDAvis: A method for visualizing and interpreting topics", *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70, 2014.

42. Bird, S., E. Klein and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, Inc., 2009.

43. Schwartz, S. H., "Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries", *Advances in Experimental Social Psychology*, Vol. 25, No. 1, pp. 1–65, 1992.

44. Suyunu, B., G. Ayci, M. Öğretir, A. T. Cemgil, S. Uskudarli, H. Zeytinoglu, B. Ozel and A. Boyacı, "Semi-Supervised Psychometric Scoring of Document Collections", *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1367–1374, IEEE, 2018.

45. Holland, J. H. *et al.*, *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT press, 1992.