A Study on Non-negative Matrix Factorization under Probability Constraints

March 2020

Hiroyoshi Ito

A Study on Non-negative Matrix Factorization under Probability Constraints

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2020

Hiroyoshi Ito

ABSTRACT

Matrix factorization, which factorizes matrix into low-rank matrices, is a central task in data analysis, such as finding hidden structure within the matrix, and matrix completion. Among matrix factorization techniques, non-negative matrix factorization (NMF), which factorizes a non-negative matrix into two non-negative matrices, has been recognized as an effective method of dimensionality reduction over matrices, because of extensibility and model explainability. NMF has been successfully applied for a wide variety of tasks in different domains, such as text, image, and audio. Several studies have attempted to give probabilistic interpretation to NMF, which brings probabilistic properties to the model, for example, the generative process of the data, and prior/posterior probability of the matrix. However, these algorithms can only be applied to limited tasks, which decays the extensibility of NMF.

In this thesis, we propose a novel framework for non-negative matrix factorization under probability constraints named probability matrix factorization (PMF), which is a novel class of matrix factorizations. In PMF, an input matrix and output matrices represent as probability matrix, in which each element represents the probability value; i.e. non-negative and the sum is 1. In this thesis, we investigate the optimization scheme for PMF, general forms of PMF, theoretical relationships between probabilistic topic models, and the applications of PMF. The optimization scheme is widely applicable to many types the differentiable loss functions such as multi-tasking problems. For the applications of PMF, we employed a probabilistic topic modeling task, and multi-tasking clustering task named CAR-clustering for multi-attributed graphs. For CAR-clustering, this thesis proposes CARPMF which consists of multiple PMFs for different tasks to ensure mutually complement each other tasks.

Experimental results showed that the superiority of PMF. Topic modeling using PMF outperforms than the ordinary NMF and LDA in terms of perplexity and clustering accuracy without losing the efficiency of NMF. CARPMF outperforms the ordinary NMF and related works of community detection tasks in terms of clustering accuracy. Moreover, CARPMF achieves better accuracy than that of do not consider probability constraints, which implies that the effectiveness of probability constrains for multi-tasking problems. These results indicate that PMF is useful for probabilistic modeling and multi-tasking problems which are prevalent in many kinds of research areas, namely data mining, machine learning, and AI researches.

Acknowledgements

This work would not have been successfully completed without help and support from many people. I would like to express my thankfulness for whom having helped me or having participated by any means to accomplish this work.

First of all, I would like to thank my supervisor Professor Toshiyuki Amagasa. This thesis has been done under the direction of him, and would not have been possible without his helpful advice and encouragement. I would like also to thank great faculty members, Professor Hiroyuki Kitagawa, Associate Professor Chiemi Watanabe, Assistant Professor Yasuhiro Hayase, Assistant Professor Hiroaki Shiokawa, and Assistant Professor Kazumasa Horie. They greatly helped and encouraged me. I would also like to express my sincere gratitude to Assistant Professor Takahiro Komamizu for his patient guidance and his strong encouragement.

I am very grateful to my doctoral committee, Professor Toshiyuki Amagasa, Professor Hiroyuki Kitagawa, Professor Mikio Yamamoto, Professor Jun Sakuma, and Associate Professor Makoto P. Kato for their valuable suggestions and constructive recommendations. Their comments greatly helped me to improve the quality of this dissertation.

I have been a research fellow of the Japan Society for the Promotion of Science (JSPS) since April 2019; This research has been supported in part by JSPS.

I would like to thank members of Kitagawa-Amagasa Data Engineering Laboratory. I am grateful to my seniors and friends, Dr. Tsubasa Takahashi, Assistant Professor Takahiro Komamizu, Dr. Yuto Yamaguchi, Dr. Yusuke Kozawa, Mr. Yuta Kusamura, Ms. Saki Nagaki, Mr. Kento Akiyama, Mr. Tomokatsu Takahashi, Mr. Yuki Sumiya, Mr. Hiroshi Yonai , Mr. Shintaro Kurimoto, and Mr. Kento Miura and for their support and goodwill. I enjoyed many discussions on research, their sharing ideas, and the collaboration of works as well as social events. Ms. Tetsuko Sato, Ms. Yumiko Hisamatsu and Ms. Shihoko Sekiya also helped me to accomplish this work.

Finally, I would like to thank my parents, Izumi and Koji, my legal father Katsuya, my younger brother Shoma, and my grandparents, Hiroshi, Shizuko, Tetsuya, Koharu, for always being supportive. Without your comprehensive supports, I would not be here.

Hiroyoshi Ito
January 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Matrix is one of the most prevalent data models to represent data that holds relationships between objects, such as social-graphs, user-item relationships in online shopping, images, and document-word cooccurrence of document datasets. For the matrix data, by applying data analysis techniques, we can detect hidden structures within the matrix or some interesting aspects of the input data.

For well understanding of such matrix, dimension reduction techniques, such as eigen-decomposition [1,2], singular value decomposition (SVD) [3] or non-negative matrix factorization (NMF) [4,5], are adapted to compress the large matrix. The basic idea of the dimension reduction techniques is approximate the input matrix by the low-dimensional matrices. These techniques enable us to understand the latent structures within the data which well represent the global tendency. These techniques are used to achieve the visualization of data, link prediction on relational data, recommendation on online shopping. Also, the low-dimensional representation of the data is used to feature or classification or clustering.

Among these techniques, NMF is very successful in understanding the latent structures in the large matrix. NMF is a method that approximates a non-negative matrix by the product of two low-rank non-negative matrices [4, 5]. Due to its efficiency and effectiveness, NMF is widely used for different types of data analysis, such as document clustering [6] and topic detection [7] over document data,

community detection over graph data [8], audio signal analysis [9], and image processing [4]. Moreover, several studies have shown that combining multiple NMF tasks could contribute to improving output quality [7, 10–13].

It should be noticed that, in many data analysis tasks, the probabilistic interpretation of models plays an important role because probability allows us to explicitly explain the generative process of the observed data, which could lead to an interpretable model generation as well. In particular, probabilistic interpretation is a quite powerful tool when combining different tasks, because it allows us to deal with completely different models in terms of probability. Moreover, by introducing the Bayesian interpretation for the model, we can define the prior and posterior distributions of the model. The prior distribution enables us to control the probability distribution of the model whereby smoothing the output probability or engage the sparseness. On the other hand, by deriving the posterior distributions of the latent variables based on the Bayes rule, we can analyze the behavior of latent variables, that enrich the model interpretability and extensibility.

Several studies have attempted to give a probabilistic interpretation to NMF [14–17]. However, these studies only focused on specific tasks, i.e. these studies can not apply to general tasks such as multitasking problems. In fact, several previous studies, where multiple NMF tasks are combined, have attempted to give an interpretation of the probability of input and output matrices [10, 11, 13]. However, the optimization methods in these studies do not pay attention to the probabilistic constraint in a matrix, i.e., a probability is a non-negative value which is less than 1 and the sum of all probabilities is equal to 1. Instead, when performing NMF, updated matrices do not meet probabilistic constraints, which are in turn forced to satisfy the constraints by applying normalization. As can easily be conjectured, it would be desirable and beneficial as well if the optimization of multiple NMF tasks can be performed in such a way that probabilistic constraints are naturally integrated.

1.2 Contributions

In this study, we propose a novel matrix factorization scheme called probability matrix factorization (PMF), which factorizes the input probability matrix into two probability matrices. The probability matrix is a matrix of which each element are interpretable as a probability value, ensuring the sum of the elements always be 1 without any operations such as normalization. This property of PMF enables

us to consider the probabilistic aspects of matrix factorization such as prediction probability, posterior probability distribution, and probabilistic generative process of the data. It is worth to notice that, the framework of PMF is widely applicable to many types the differentiable loss functions, which enables us to consider wide variety of loss measures and complex forms of loss functions such as multitasking problems. This thesis investigates how to realize PMF, and applicability of PMF to data mining tasks. This section summarizes contributions of this thesis.

1.2.1 Framework for Probability Matrix Factorization

In this study, we provide a framework for probability matrix factorization (PMF). PMF is defined as a minimization problem for loss function under constraints. The loss function measures the difference between the input probability matrix and the multiplication of the output matrices. Constraints for the output matrices consist of non-negativity constraints which constrains each element to have non-negative value and equality constraint enforces that the sum of the elements is 1 to ensure the element represents the probability.

In this thesis, we investigate the optimization scheme for PMF. We derive iterative updating rules for the output matrices based on Karush-Kuhn-Tucker (KKT) conditions of non-negativity and equality constraints for probability interpretation of the output matrices. The derived optimization scheme is applicable to a wide range of differentiable loss functions. We theoretically prove that the updating rule monotonically decreases the loss function and the output matrices always meet the constraints while not sacrificing the calculation cost.

For more general use of PMF, we derive the variation of the PMF. Specifically, we investigate the patterns of probability constraints and regularization for output matrices. As for probability constraints, we derive 4 patterns of constraints that probabilistic interpretation of elements in the input matrix and of multiplication of the output matrices are consistent. As for the regularization term, we introduce Dirichlet regularization, which controls the probability distribution of the output matrices.

Moreover, we investigate the relationships between PMF and probabilistic topic models by analyzing the optimization problem of PMF and the other topic models. As a result, we found that the special cases of PMF strongly relates to pLSA and LDA.

Experiments using benchmark document datasets revealed that PMF outperforms the ordinal NMF and LDA in the viewpoints of perplexity and document clustering while the calculation time is compatible with the ordinal NMF.

1.2.2 Multi-task Probability Matrix Factorization

PMF facilitates to mixture multiple tasks more precisely, because of the probability values in PMF explicitly describes the probabilistic generative processes of the data. In this study, we examine the usefulness of PMF for a multi-tasking model.

The task is named CAR-clustering, which is a clustering scheme for multi-attributed graphs in which the nodes have multiple types of attributes. CAR-clustering includes **C**ommunity detection over nodes of graphs, **A**ttribute-cluster detection for multi-types of attributes and the **R**elationships detection between communities and attribute-clusters. By detecting them, the community is characterized by the related attribute clusters, and the relationships between different communities are also described by the related attribute clusters. CAR brings us the global view of large scale multi-attributed graphs.

For this task, we propose a model for CAR-clustering called CARPMF, where the loss function consists of multiple types of PMF loss functions to make the loss functions mutually complement each other. The basic idea of CARNMF is that communities and the attribute clusters are strongly related to each other. This assumption is supported by the property of graphs called homophily effects [18].

Experiments using real-world bibliographic datasets show that CARNMF performs better than the ordinary NMF, related works, and CARPMF that do not consider the probabilistic constraints, in the viewpoint of the accuracy of community detection and attribute cluster detection. Moreover, CARPMF detects reasonable communities, attribute clusters and the relationships between communities and the attribute clusters.

1.3 Overview of the Thesis

In the previous section, we discuss background, motivations and main contributions. This section summarizes the overview of the thesis:

Chapter 2: Background and Survey. In this chapter, we discuss the position of this thesis by thoroughly reviewing the related works. This chapter summarizes relationships between PMF and traditional matrix factorization techniques, applications of NMF which is the range of the applications of PMF, related works of CARPMF, relationships between optimization schemes for PMF and NMF.

Chapter 3: A Framework of Probability Matrix Factorization. This chapter provides a framework for PMF. We formally define the optimization problem for PMF and derive an optimization scheme for PMF and give a generalized form of PMF. This chapter also describes the relationships between PMF and the probabilistic topic models. The experiments using benchmark datasets show that PMF outperforms ordinary NMF and LDA in the viewpoint of perplexity and clustering accuracy.

Chapter 4: A Multi-tasking Probability Matrix Factorization. In this chapter, we investigate the usefulness of PMF for a multi-tasking model. This chapter formally defines the multi-attributed graphs, CAR-clustering. Then, CARPMF is proposed as a mixture of multiple PMFs for CAR-clustering. Experimental results using real-world bibliographic datasets show that CARPMF outperforms ordinary NMF and related works.

Chapter 5: Conclusions. This chapter concludes the thesis and outlines some directions for future advances in this research.

Chapter 2

Background and Survey

As mentioned in chapter 1, in this thesis, we investigate non-negative matrix factorization under probability constraints which is called probability matrix factorization (PMF) of which the input and output matrices are interpreted as probability matrix. This chapter, we discuss the position of this research by thoroughly reviewing the related works. Section 2.1 discusses relationships between NMF and another matrix factorization techniques. Section 2.2 reviews the possible applications of NMF. Section 2.3 gives related algorithms for optimizing the loss function of NMF. Section 2.4 reviews the related works that attempted to give probabilistic interpretation to NMF.

2.1 Matrix Factorization

There are many kinds of studies which decomposes input matrix into low rank matrices. NMF is a special case of the matrix factorization techniques. In the following subsection, we summarize the variants of the matrix factorizations, namely eigen-decomposition, singular value decomposition, and non-negative matrix factorization, which are the most fundamental techniques in the area of matrix factorization.

2.1.1 Eigen-decomposition

Eigen-decomposition [1, 2] decomposes input real-valued square matrix into two eigenvector matrix and eigenvalue matrix. Given a input matrix $A \in \mathbb{R}^{N \times N}$ the eigen-decomposition is:

$$A = P \Lambda P^{-1}, \tag{2.1}$$

where $P \in \mathbb{R}^{N \times N}$ is eigenvector matrix whose column represents eigenvector, and $\Lambda \in \mathbb{R}^{N \times N}$ is a diagonal matrix, which is called eigenvalue matrix, whose elements contains eigenvalues.

Eigen-decomposition have many possible applications: minimum cut in graph data, graph matching [19], spectral clustering [20] and kernel k-means clustering [21].

2.1.2 Singular Value Decomposition

One of the most famous method of matrix decomposition technique is singular value decomposition (SVD) [3], which factorizes the input matrix into two orthogonal matrices and a singular value matrix. Given a matrix $X \in \mathbb{R}^{N \times M}$, singular value decomposition is:

$$X = U \Sigma V^{\top}, \tag{2.2}$$

where Σ is a diagonal matrix called singular value matrix whose element contains the singular values, and $U \in \mathbb{R}^{N \times N}$ and $V \in \mathbb{R}^{M \times M}$ are the eigenvector matrix of $XX^{\top}$ and $X^{\top}X$, respectively.

SVD enables us to get low rank representation by ignoring the lowest singular values. It is well known that principal component analysis (PCA) [22] is identical to SVD. Latent semantic analysis [23] for text data is very famous approach for dimension reduction based information retrieval. SVD have many potential applications such as recommender system [24, 25], graph embedding [26, 27], and word embeddings [28].

2.1.3 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) [4, 5] decomposes input non-negative matrix into two non-negative matrices. Given a non-negative matrix $X \in \mathbb{R}_{+}^{M \times N}$, NMF is:

$$X \approx UV, \tag{2.3}$$

where $U \in \mathbb{R}_{+}^{M \times K}$ and $V \in \mathbb{R}_{+}^{K \times N}$ are the output matrices. K is usually set to $K \ll M, N$ The output matrices are found by minimizing the loss function which measures the difference between input matrix and product of the output matrices. Thanks to the output matrices always satisfy non-negative value, NMF has high model interpretability than the other matrix factorization techniques. This nature of NMF enables us to develop many kinds of applications.

2.2 Applications of Non-negative Matrix Factorization

Non-negative Matrix Factorization is one of the most famous data mining techniques in the context of data summarization, recommendation and dimension reduction. In this section, we discuss the applications of NMF and relationships between our proposals and the other methods. Especially, we focus on the representative categories below:

1. **Topic modeling.**
2. **Community detection.**
3. **Recommendation.**
4. **Audio signal processing.**
5. **Multi-tasking method.**

2.2.1 Topic Modeling

In this thesis, we evaluate the effectiveness of PMF in the task of topic modeling. In this section, we review related works on topic modeling. Topic modeling is a task summarizing documents of natural languages. Typicaly, topic models assume each document has topic proportions and each topic has the word proportions. By maximizing the appearance probability words, the model expresses the bird-eye-view of the documents.

The primal studies on topic modeling are probabilistic latent semantic analysis (p-LSA) [29] and latent Dirichlet allocation (LDA) [30]. p-LSA parameterize the topic proportions of each of the input documents and the word proportion of each of the topics. The parameters are optimized by using the EM algorithm [31]. LDA is a first full-bayes model for topic modeling, which infers the probability of word occurrence in each document by integrate out all of the parameters. Note that, by assuming the prior probability distribution to models in p-LSA, namely the topic proportion and the wor proportion, the model represents as LDA. For inference, there are several schemes to maximize the likelihood of the output words, namely variational bayes inference [30, 32] and gibbs sampling [33, 34].

There are many variants of topic models. CTM [35] models the relationships between topics via introducing a matrix that represents the relationships between topics. DTM [36] models the chronological change of the topic distributions. HDP [37] introduces Dirichlet process for the model that can deal with the infinite mixture of the topics and can determine the proper number of topics. TM-LDA [38] predicts the future proportion of the topics.

NMF is also used for topic detection, especially in the context of information summarization. By applying NMF to the document-term relation matrix, we can detect topics in the document and the topic weight of each document. In this scenario, one output matrix represents the word weight of each topic while another output matrix represents the topic weights for each of the documents. Several studies [6, 39, 40] proposed a mathod that detects topics from documents via NMF. Cao et al. [41] proposed online NMF which track the change of the latent factors, and proposed the algorthm that efficiently learns the latent factors. Kasiviswanathan et al. [42] proposed a method that detects emerging topics from text streams. Vaca et al. [43] proposed a topic tracking method based on NMF to monitor the evolution of topics. Choo et al. [44] proposed a NMF based document visualization system that interactively detect topics and visualize the document. Saha et al. [45] proposed

a method that detect, track and emergence of topics via a temporal regularization term. Kim et al. [46] proposed a method that simultaneously detects common topics and discriminative topics from two document datasets.

For evaluating topic models, perplexity is a major criterion that measures the prediction likelihood of word appearance on unseen documents. Even though NMF is useful for document clustering and summarization, NMF can not be applied to predict word occurrence. It is because of NMF is not a probabilistic model. Our proposal named probabilistic matrix factorization overcome the shortcomings of NMF by bringing strict probabilistic interpretation for NMF based modeling.

2.2.2 Community Detection

Community detection is a task that detects densely connected nodes from graphs. NMF is one of the useful tools for detecting overlapping communities. By applying NMF to an adjacency matrix of a graph, node-community relation matrices are detected. By interpreting each element of the matrix as membership of a community, we can achieve community detection.

In this thesis, we propose multi-tasking PMF that contains community detection. In this section, we summarize the related works that attempted to detect the communities in graph, and discuss the relationships between the related works.

Community detection in graphs is a current topic of interest in graph analysis and AI research. Existing works for non-attributed graphs can be categorized according to the techniques used: graph separation [47, 48], probabilistic generative model [49–51], and matrix factorization [52–54]. [47] defined *modularity*, which indicates how separated a community is from other nodes. More comprehensive surveys can be found in [55, 56].

Recently, several works have addressed the problem of detecting communities and their semantic descriptions on node-attributed graphs. [57] proposed *CESNA*, where communities and their attributes are simultaneously detected in an efficient manner. [58] proposed *SCI* to detect communities and their semantics using NMF. [59] proposed a probabilistic generative model called the *author-topic model* to model communities and related topics. [60] proposed *COMODO* to detect communities with shared properties using subgroup discovery techniques. [61] proposed a method for detecting communities and their descriptions from an attributed graph

where detection of communities and induction of description are alternated. [62] proposed a joint community profiling and detection model which characterizes communities with user published contents and user diffusion links. Likewise, [63] proposed *LCTA*, where communities and their topics are modeled separately, and then their relationships are modeled using a probabilistic generative model. A comprehensive survey over these works can be found in [64].

The aforementioned works only consider single textual attributes or uniformly handle multiple attributes without any distinction. In reality, each attribute represents different aspects of the nodes. In our research, we deal with heterogeneous attributes individually. In addition to community detection, we perform clustering over attribute values for each attribute, which, in turn, can be used to improve the quality of communities detected.

Some works have investigated clustering over networks containing different types of nodes and/or edges. [65] studied community detection with characterization from multidimensional networks, which is defined as a graph consisting of a set of nodes and multiple types of edges. [66] studied subgraph detection from multilayer graphs with edge labels. In contrast, we assume a different model where each node is characterized by multiple attributes. As we shall see later, we model multiple attributes using different types of nodes, and community detection as well as attribute-value clusterings can be described on such a graph consisting of different types of nodes (nodes and multiple types of attribute values), and try to detect communities over the nodes as well as the clusters over other types of attribute values. [67] proposed a scheme of ranking-based clustering for multi-typed heterogeneous networks, where two or more types of nodes are included. Similarly, [11] proposed an NMF-based method for such networks. These methods differ from ours in that they define a cluster consisting of all types of nodes. In other words, these methods cannot handle each attribute in a unique way. In contrast, our work deals with different attributes individually, but solves community detection and attribute-value clustering in a unified manner.

More recently, several kinds of network methods are proposed, which can cluster nodes/attributes by applying a vector space based clustering methods for vector representations of nodes/attributes. Deepwalk [68] achieves vector representation of nodes by applying skip-gram [69] for the sequences of nodes by sampled short random walks. LINE [70] learns vector representations of nodes by preserving one-hop and two-hop adjacency of nodes. node2vec [71] extends DeepWalk with a controlled path sampling process, which captures proximity and structural similarity of

nodes on a graph. LANE [72] is a method that captures the vector representations of nodes which consider proximity of nodes, attribute similarity and label similarity. metapath2vec [73] achieves vector representations of any kinds of nodes in heterogeneous information networks, by applying skip-gram for sequences of nodes which controlled by meta-path schemes [74]. These methods are research aimed at network embedding, fundamentally different from the purpose of our research.

2.2.3 Recommendation

Recommendation is a task that recommend items to the users in a system. A basic idea of the recommendation by matrix factorization is matrix completion via multiplication of low-rank matrices derived by matrix factorization [75, 76]. By using NMF for user-item relationship matrix with missing values, the output matrices captures the hidden preferences of the users and the hidden properties items, and by reconstructing the input matrix, the missing values are filled in, i.e. system can recommend the items to the users. Zhang et al. [77] proposed NMF based recommendation system which fills user rating for items in an incomplete rating matrix. Luo et al. [78] proposed a fast algorithm for NMF based collaborative filtering. Yoo et al. [79] proposed weighted non-negative matrix co-tri-factorization which collaboratively predicts the rating with side information of contents. Li et al. [80] proposed NMF based privacy preserving recommendation algorithm.

2.2.4 Audio Signal Processing

NMF is also used for signal audio processing. In this scenario, the input matrix typically represents relationships between frequency bins and time frames. By factorizing the matrix, different sources of the signals are detected. Wilson et al. [81] proposed NMF based method that separates the input signal into clean sound and noises by modeling the probability distribution of the sound and the noises. Cichocki et al. [82] proposed an NMF based blind source separation method, that separates the sound data into sources of sounds. Wang et al. [83] proposed a procedure for the separation of pitched musical instruments and drums from polyphonic music using NMF. Ozerov et al. [84] proposed a method for detecting sound source from multi-channel audio data.

2.2.5 Multi-Tasking Method

s For more complicated tasks, several studies have shown that concurrent processing of two or more NMF tasks is useful to improve the output performance. Wang et al. [10] showed that, by solving NMF of community detection and detecting attributes of communities from an attributed network, the quality of detected community could be improved. Liu et al. [12] proposed multi-view clustering scheme based on multiple NMFs for multi-view data. In our previous work, we proposed an NMF based clustering scheme for multi-attributed graph called CARNMF [13], which outputs community detection, attribute-value clustering, and derivation of relationships between communities and attribute-value clusters at the same time. Notice that all of these researches assume probabilistic interpretation on the output matrices, thereby enabling integration of different types of data. However, none of the above studies guarantee that output matrices satisfy the probability constraints and/or the optimization method is exclusively dedicated for the problem being addressed in the respective paper and is therefore not applicable to other problems.

2.3 Algorithms for Non-negative Matrix Factorization

In this section, we review the algorithms in previous studies for optimizing the NMF, and discuss the relationships between algorithm for PMF and them.

To realize NMF, we consider the loss function which measures deference between input matrix and output matrices. NMF is accomplished by minimizing the loss function under non-negative constraints. Given an input non-negative matrix $X \in \mathbb{R}_{+}^{M \times N}$, typical example of loss function for NMF is as follows:

$$U, V = \arg\min_{U,V} \; L = \arg\min_{U,V} \; \mathcal{D}\left(X||UV\right) \\ subject\ to \;\; U \geq 0, \;\; V \geq 0. \tag{2.4}$$

$\mathcal{D}$ is any distant metrics for matrices, typical examples are Frobenius norm and

generalized KL-divergence, where

$$\mathcal{D}_{Fro}(X||UV) = \|X - UV\|_F^2 \tag{2.5}$$

$$\mathcal{D}_{KL}(X||UV) = \sum_{d,w} X_{dw} \log \frac{X_{dw}}{U_{d\cdot}V_{\cdot w}} - X_{dw} + U_{d\cdot}V_{\cdot w}, \tag{2.6}$$

respectively.

For this optimization problem, there are several researches that attempted to optimize the loss function. Lee et al. [5] proposed a optimization algorithm which consists of alternative updating rules for each of the output matrix. Each of the updating rules are composed of multiplication of the matrices, which is called multiplicative updating rules. The updating rules are following:

$$U'_{dz} \leftarrow U_{dz} \frac{[\partial_{U_{dz}} L]^-}{[\partial_{U_{dz}} L]^+}, \quad V'_{zw} \leftarrow V_{zw} \frac{[\partial_{V_{zw}} L]^-}{[\partial_{V_{zw}} L]^+},$$

where $[\partial_{U_{dz}} L]^+$ and $[\partial_{U_{dz}} L]^-$ are the plus part and the minus part of the partial differential function, respectively. The optimization process is derived by KKT conditions [85–87] for the optimization problem. Under the algorithm, the loss function is non-increase, and at the convergence point, it is ensured that the loss function is the local minimum. The optimization algorithm proposed in chapter 3 is based on the multiplicative updating rules.

There are variants of optimization algorithms for NMF. Hoyer [88] proposed an optimization method based on gradient descent. Under the algorithm, the minus values are projected to 0 to ensure the non-negativity. However, there is a drawback that the projection of the algorithm makes harder to analyze the convergence of the loss function. Paatero and Tapper [89] proposed an algorithm based on the alternative least square (ALS) algorithm. The basic idea of ALS based algorithm is that alternately find the optimal values of the output matrices. Because of the loss function of NMF is a convex function when assuming an output matrix is a constant, when the gradient for the objective matrix is 0, the point would be the optimal value. So that, by alternately find the optimal value, the loss function for NMF is gradually decreased.

2.4 Probabilistic Interpretation for Matrix Factorization

This section discusses the related works on probabilistic interpretation of matrix factorization techniques. Specifically, we discuss in two different viewpoints; (1) probabilistic interpretation for non-negative matrix factorization, (2) Bayesian modeling for matrix factorization.

2.4.1 Probabilistic Interpretation for Non-negative Matrix Factorization

Several works have studied how NMF can be applied to probabilistic models. Studies in Ding et al. [14, 15], and Gaussier and Goutte [90] revealed that the loss function of NMF under KL-divergence is equivalent to the loss function of pLSI. By contrast, in this thesis, we investigate the optimization scheme for not only the KL-divergence based but the Frobenius norm, which is more prevalent in NMF based research. Moreover, the optimization scheme of ours can deal with more general form of non-negative matrix factorization that has a differentiable loss function.

Luo et al. [16] proposed probabilistic NMF for topic modeling that directly approximates the input probabilistic matrix by the product of output probabilistic matrices with low rank. However, the proposed optimization algorithm needs iterative updating for each output matrix in addition to the outer iteration, which makes the number of calculation much larger than the ordinary NMF. By contrast, as we will see later, the computational complexity of our proposed optimization scheme the same to the ordinary NMF and also the learning speed is comparable. Moreover, our optimization scheme is widely applicable to other problems that exploit probabilistic matrix.

2.4.2 Beyesian Modeling for Matrix Factorization

Several studies have attempted to introduce bayesian interpretation for matrix factorization. Mnih et al. [91] proposed a probabilistic matrix factorization of which the elements in the output matrix represents the probabilistic random values that

have prior distributions. By assuming the prior distribution, we can avoid overfitting to the output matrix. The output matrices are estimated by MAP estimation. Salakhutdinov and Mnih [92] proposed a fully Bayesian treatment for probabilistic matrix factorization that can deal with the uncertainty of the matrix by integrating over all parameters and the hyperparameters. It has been revealed that bayesian matrix factorization is effective to many kinds of data mining tasks such as link prediction [76, 93], predicting drug-target interaction [94, 95], and recommendation [93, 96–99]. However, because of these models assume that elements in the output matrix are random variables of the normal distribution, the elements represent not only the positive variables but the negative variables.

Schmidt [17] proposed Bayesian non-negative matrix factorization, that of the elements in the output matrices are assumed to be generated by gamma distribution as prior distributions. The Bayesian NMF is widely used for data mining tasks [54, 100, 101] because that can avoid overfitting to the input data. However, in these models, the elements are merely the random variables, that do not represent the probability value that describes the generative process of the elements. By contrast, in this study, we investigate the matrix factorization technique of which the matrices represent the probability matrix; i.e. each element represents the probability value. So that the probabilistic generative process of the data is expressed in each element in the matrix, that enrich the model explainability, and enables us to collaborate the multiple tasks.

Chapter 3

A Framework for Probability Matrix Factorization

In this chapter, we propose a novel matrix factorization technique called probability matrix factorization (PMF). PMF factorizes the input probability matrix into two probability matrices. PMF is formulated as a minimization problem of loss-function under inequality condition (non-negativity) and equality constraints (the sum of the elements equals 1). For PMF, we develop an iterative updating algorithm that finds the local minimum of the loss function satisfying the conditions while do not sacrifice the efficiency. The algorithm can be applied to a wide range of differentiable loss-functions of which nature enables us to consider many kinds of tasks. Moreover, theoretical analyses for the loss-function reveal the relationships between PMF and the probabilistic topic models namely pLSA and LDA. The experiments using benchmark document datasets show that PMF achieves higher performance than the related works in topic modeling in the viewpoint of perplexity and topic-based document clustering. This chapter gives a formal definition of PMF, optimization algorithm, extensions of PMF, theoretical analysis for loss-function of PMF, and experiments for topic modeling using PMF.

3.1 Introduction

In this thesis, we propose a novel matrix factorization scheme called probability matrix factorization (PMF). We develop a novel optimization method for NMF under probabilistic constraint, which is called probability matrix factorization, whereby the output matrices are interpretable as probability matrices without any operation such as normalization. Specifically, we exploit topic modeling as an example of PMF and derive an optimization scheme for it. The loss function is formulated as difference between *document-term* matrix (each element indicates the joint probability of presence of a document and a term) and multiplication of *document-topic* matrix (each element indicates the joint probability of presence of a document and a topic) and *topic-term* matrix (each element indicates the conditional probability of presence of a term given a topic). We derive iterative updating rules for the output matrices based on Karush-Kuhn-Tucker (KKT) conditions [85–87] of non-negativity and equality constraints for probability interpretation of the output matrices.

For more general use of PMF, we extend PMF to a more flexible formulation. In this formulation, loss measure between the input and the outputs are generalized, and regularization terms for the output matrices are taken into account. For probability constraint, we derived 4 variety of PMFs, which the probability interpretation of the input matrix and the product of the output matrices are consistent. For the regularization term, we propose a novel regularization term called Dirichlet regularization term, as a log-likelihood of Dirichlet distribution of the output matrix, which enables us to control the output matrices to smooth or sparse.

Moreover, we theoretically analyze the loss-function of PMF. Consequently, we revealed that PMF strongly relates to probabilistic topic models. Specifically, (1) minimizing the loss-function of PMF of the third variation in generalized KL-divergence is equivalent to maximizing the log-likelihood of probabilistic latent semantic analysis (pLSA) , and (2) minimizing the loss-function of PMF of the first variation in generalized KL-divergence with Dirichlet regularization is equivalent to MAP estimation of latent Dirichle allocation (LDA) in which the hidden variable z is marginalized out.

The key contributions of this research can be summarized as follows:

- We propose a novel matrix factorization scheme, probability matrix factorization (PMF), which factorized the input probability matrix into two prob-

ability matrices.

- We propose a novel optimization scheme for PMF. For the method, we give theoretical supports for the validity of update rules: (1) our optimization scheme monotonically decrease the objective function; (2) the output matrices always meet the probability constraints; (3) the time complexity of the proposed optimization scheme remains the same as the ordinary NMF; (4) our optimization scheme can apply to many kind of differentiable loss-functions.

- We propose a novel regularization term for the output matrices, Dirichlet regularization term, which enables us to control the probability distribution of the output matrix.

- The theoretical analysis for loss-function of PMF revealed that PMF strongly relates to probabilistic topic models, namely pLSA [29] and LDA [30].

- The experimental results show that PMF is more accurate than the ordinary NMF regarding clustering and perplexity without sacrificing efficiency compared with ordinary NMF.

The rest of this chapter is organized as follows. Section 3.2 states the problem definition of PMF. Section 3.3 describes the derivation process of an optimization scheme for PMF. In Section 3.4, we extend PMF to more general forms and introduce the Dirichlet regularization. In Section 3.5, we analyze the loss-function of PMF and derive the relationships between probabilistic topic models. Section 3.6 provides experiments for topic modeling using benchmark document datasets. Finally, Section 3.7 concludes this chapter.

3.2 Problem Definition

In this study, we propose an optimization scheme for non-negative matrix factorization (NMF) under probabilistic constraints where each output matrix is constrained to be stochastic; i.e., each element is a non-negative real number representing probability. Hereafter, we call it probability matrix factorization (PMF). In this paper, we exploit topic modeling over documents as an easy-to-understand example to derive optimization for loss function. However, the proposed scheme is not limited

to topic modeling and can be applied to other problems as long as it is based on probabilistic representations of matrices.

First, we give a formulation of loss function for topic modeling under PMF. Figure 3.1 is an image of probability matrix factorization. Given a set of documents, we represent them using a document-term matrix $X \in \mathbb{R}^{N \times M}$, where N is the number of documents and M is the length of a dictionary. We normalize X as $\sum_{d,w} X_{d,w} = 1$ so that each element $X_{d,w}$ in matrix X represents the joint probability $p(d, w)$ of document d and word w in the input document. In the PMF based topic modeling, the ratio of the topic of the document is represented by a matrix $U \in \mathbb{R}^{N \times K}$, where K is the number of topics satisfying $K \ll M$. The rows and columns of matrix U denote documents and topics, respectively. Each element $U_{d,z}$ of the matrix represents the joint probability $p(d, z)$ of document d and topic z. Also, the probability distribution of words in the topic is represented by a matrix $V \in \mathbb{R}^{K \times M}$. Similarly, the rows and columns of matrix V denote topics and words, respectively. Each element $V_{z,w}$ of the matrix represents the conditional probability $p(w|z)$, i.e., the appearance of word w given topic z. In probability $p(d, z, w)$, document d has word w through topic z which is represented by $U_{d,z}V_{z,w} = p(d, z)p(w|z) = p(d, z, w)$. Furthermore, the joint probability $p(d, w)$, which is equal to $X_{d,w}$, is expressed by $\sum_z U_{d,z}V_{z,w} = \sum_z p(d, z, w) = p(d, w)$. By using frobenius norm as a function to evaluate the difference, the optimization problem is expressed as follows:

$$
\begin{aligned}
U, V = \ & \underset{U,V}{\arg\min} \ \|X - UV\|_F^2 \\
\textit{subject to} \ & U \geq 0, \qquad\qquad V \geq 0, \\
& \sum_{d,z} U_{dz} = 1, \qquad \sum_{w} V_{zw} = 1, \ \forall z.
\end{aligned}
\tag{3.1}
$$

This optimization problem is formulated as a minimization problem under two types of the constraints: inequality constraints as non-negativity constraints, equality constraints as the probability constraints. By solving this problem, the most reasonable topic model can be estimated, thereby making it possible for us to get an interpretation of the probability to the output matrices U and V.

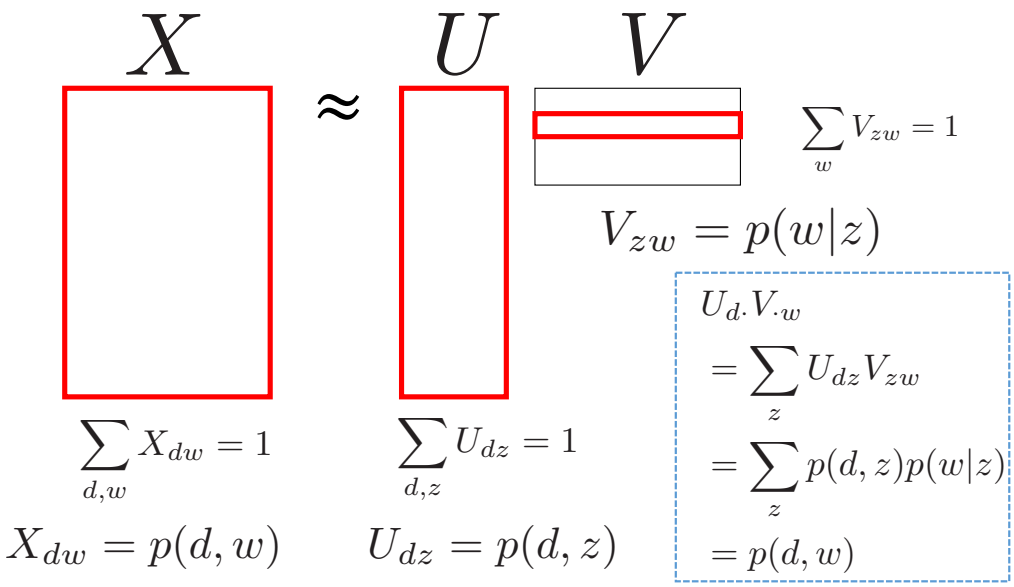


Figure 3.1: **An image of probability matrix factorization.** Each element of matrices represents the probability value. The sum of the elements in red rectangle is 1. The probabilistic interpretation of elements in the input matrix and multiplication of the output matrices is consistent; i.e. both are $p(d, w)$.

3.3 Optimization

In this section, we propose an optimization scheme for the problem described in the previous section. Similar to the ordinary NMF, the loss function L is convex with respect to each output matrix (U or V) but is not when considering both output matrices at the same time. In this paper, we propose iterative update rules for U and V whereby the output matrices are alternately adjusted while optimizing the loss function L. However, unlike the ordinary NMF, our PMF has equality constraints as the probability constraints on U and V, leading to a more complicated optimization problem. To tackle this problem, we employ Karush-Kuhn-Tucker (KKT) condition to derive the update rules for each matrix U and V. In the KKT conditions, the Lagrange multipliers are introduced for the equality constraints in addition to the non-negativity constraints, which must be determined. As a result, by determining the Lagrange multipliers for non-negativity and equality constraints, we derive the following update rules:

$$U'_{dz} \leftarrow U_{dz} \frac{[\partial_{U_{dz}} L]^- + \lambda^-}{[\partial_{U_{dz}} L]^+ + \lambda^+}, \tag{3.2}$$

$$\lambda^- = \frac{1 - \sum_{d,z} \frac{U_{dz}\left[\partial_{U_{dz}} L\right]^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+}}{\sum_{d,z} \frac{U_{dz}}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+}}, \tag{3.3}$$

$$\lambda^+ = \max\left(\max\left(-\partial_{U_{\cdot,\cdot}} L\right), 0\right), \tag{3.4}$$

$$V'_{zw} \leftarrow V_{zw} \frac{[\partial_{V_{zw}} L]^- + \psi_z^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+}, \tag{3.5}$$

$$\psi_z^- = \frac{1 - \sum_w \frac{V_{zw}[\partial_{V_{zw}} L]^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+}}{\sum_w \frac{V_{zw}}{[\partial_{V_{zw}} L]^+ + \psi_z^+}}, \tag{3.6}$$

$$\psi_z^+ = \max\left(\max\left(-\partial_{V_{z\cdot}} L\right), 0\right), \tag{3.7}$$

where

$$\begin{aligned}
\partial_{U_{dz}} L &= [\partial_{U_{dz}} L]^+ - [\partial_{U_{dz}} L]^-, \\
&[\partial_{U_{dz}} L]^- = (XV^\top)_{dz}, \ [\partial_{U_{dz}} L]^+ = (UVV^\top)_{dz}, \\
\partial_{V_{zw}} L &= [\partial_{V_{zw}} L]^+ - [\partial_{V_{zw}} L]^-, \\
&[\partial_{V_{zw}} L]^- = (U^\top X)_{zw}, \ \ [\partial_{V_{zw}} L]^+ = (U^\top UV)_{zw}.
\end{aligned}$$

These update rules monotonically decrease the loss function under satisfying the constraints. From the following subsections, we will describe the derivation process of each update rule and theoretical supports of validity about update rules.

3.3.1 Derivation of update rules

Based on KKT condition, we introduce Lagrange multiplier matrices $\Theta = (\theta_{dz})$ and $\Xi = (\xi_{zw})$ for non-negative constraint of U_{dz} and V_{zw}, respectively, and Lagrange multipliers λ and $\Psi = (\psi_i)$ for equality constraint of U_{dz} and V_{zw}, respectively. We

have the following equivalent objective functions:

$$\mathcal{L}(U_{dz}) = L + \sum_{d,z} \theta_{dz} U_{dz} + \lambda \left(\sum_{d,z} U_{dz} - 1 \right), \tag{3.8}$$

$$\mathcal{L}(V_{zw}) = L + \sum_{z,w} \xi_{zw} V_{zw} + \sum_{z} \psi_z \left(\sum_{w} V_{zw} - 1 \right). \tag{3.9}$$

KKT conditions are:

$$U \geq 0,\ V \geq 0,\ \sum_{d,z} U_{dz} = 0,\ \sum_{w} V_{zw} = 0,\ \forall k.\ (Primal\ feasibility) \tag{3.10}$$

$$\Theta \geq 0,\ \Xi \geq 0.\ (Dual\ feasibility) \tag{3.11}$$

$$U_{dz}\theta_{dz} = 0,\ \forall d, z,\ V_{zw}\xi_{zw} = 0,\ \forall z, w.\ (Complementary\ slackness) \tag{3.12}$$

Set derivative of $\mathcal{L}(U_{dz})$ with respect to U_{dz}, and $\mathcal{L}(V_{zw})$ with respect to V_{zw} to 0, we have:

$$\partial_{U_{dz}} \mathcal{L}(U_{dz}) = \partial_{U_{dz}} L + \theta_{dz} + \lambda = 0, \tag{3.13}$$

$$\partial_{V_{zw}} \mathcal{L}(V_{zw}) = \partial_{V_{zw}} L + \xi_{zw} + \psi_z = 0. \tag{3.14}$$

Following the KKT condition for the non-negativity of U_{dz}, and V_{zw}, we have the following equations:

$$U_{dz}\theta_{dz} = U_{dz}\left(-\partial_{U_{dz}} L - \lambda\right) = 0, \tag{3.15}$$

$$V_{zw}\xi_{zw} = V_{zw}\left(-\partial_{V_{zw}} L - \psi_z\right) = 0. \tag{3.16}$$

Let $\partial_{U_{dz}} L = [\partial_{U_{dz}} L]^+ - [\partial_{U_{dz}} L]^-$ and $\lambda = \lambda^+ - \lambda^-$, where $[\partial_{U_{dz}} L]^+ \geq 0$, $[\partial_{U_{dz}} L]^- \geq 0$, $\lambda^+ \geq 0$, and $\lambda^- \geq 0$, the equation 3.15 is equivalent to:

$$U_{dz}\left([\partial_{U_{dz}} L]^+ - [\partial_{U_{dz}} L]^- + \lambda^+ - \lambda^-\right) = 0. \tag{3.17}$$

Similarly, let $\partial_{V_{zw}} L = [\partial_{V_{zw}} L]^+ - [\partial_{V_{zw}} L]^-$ and $\psi_z = \psi_z^+ - \psi_z^-$, where $[\partial_{V_{zw}} L]^+ \geq 0$, $[\partial_{V_{zw}} L]^- \geq 0$, $\psi_z^+ \geq 0$, and $\psi_z^- \geq 0$, the equation 3.16 is equivalent to:

$$V_{zw}\left([\partial_{V_{zw}} L]^+ - [\partial_{V_{zw}} L]^- + \psi_z^+ - \psi_z^-\right) = 0. \tag{3.18}$$

These are the fixed point equations that the solutions must satisfy at convergence. Given an initial value of U_{dz} and V_{zw}, we can get the following update rules:

$$U'_{dz} \leftarrow U_{dz} \frac{[\partial_{U_{dz}} L]^- + \lambda^-}{[\partial_{U_{dz}} L]^+ + \lambda^+}, \tag{3.19}$$

$$V'_{zw} \leftarrow V_{zw} \frac{[\partial_{V_{zw}} L]^- + \psi_z^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+}. \tag{3.20}$$

Next, we determine the Lagrange multipliers λ and ψ_z for equality constraints $\sum_{d,w} U_{dz} = 1$ and $\sum_w V_{zw} = 1$. λ^+, λ^-, ψ_i^+, and ψ_i^- must satisfy following equations:

$$\sum_{d,z} U'_{dz} = \sum_{d,z} U_{dz} \frac{[\partial_{U_{dz}} L]^- + \lambda^-}{[\partial_{U_{dz}} L]^+ + \lambda^+} = 1, \tag{3.21}$$

$$\sum_{w} V'_{zw} = \sum_{w} V_{zw} \frac{[\partial_{V_{zw}} L]^- + \psi_z^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+} = 1 \tag{3.22}$$

which are equivalent to:

$$\sum_{d,z} \frac{U_{dz} [\partial_{U_{dz}} L]^-}{[\partial_{U_{dz}} L]^+ + \lambda^+} + \lambda^- \sum_{d,z} \frac{U_{dz}}{[\partial_{U_{dz}} L]^+ + \lambda^+} = 1, \tag{3.23}$$

$$\sum_{w} \frac{V_{zw} [\partial_{V_{zw}} L]^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+} + \psi_z^+ \sum_{w} \frac{V_{zw}}{[\partial_{V_{zw}} L]^+ + \psi_z^+} = 1. \tag{3.24}$$

From Equations 3.23 and 3.24, let λ^+ and ψ_z^+ be the parameters of λ^- and ψ_z^-, respectively. With the constraints $\lambda^+, \lambda^-, \psi_z^+, \psi_z^- \geq 0$, we have the following inequalities:

$$\lambda^- = \frac{1 - \sum_{d,z} \frac{U_{dz} \left[\partial_{U_{dz}} L\right]^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+}}{\sum_{d,z} \frac{U_{dz}}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+}} \geq 0, \tag{3.25}$$

$$\lambda^+ \geq 0, \tag{3.26}$$

$$\psi_z^- = \frac{1 - \sum_{w} \frac{V_{zw} [\partial_{V_{zw}} L]^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+}}{\sum_{w} \frac{V_{zw}}{[\partial_{V_{zw}} L]^+ + \psi_z^+}} \geq 0, \tag{3.27}$$

$$\psi_z^+ \geq 0. \tag{3.28}$$

Under satisfying inequalities 3.25 and 3.26, the updated U'_{dz} satisfies $\sum_{d,w} U'_{dz} = 1$,$U'_{dz} \geq 0$; and satisfying inequalities 3.27 and 3.28, the updated V'_{zw} satisfies $\sum_{w} V_{zw} = 1$,$V'_{zw} \geq 0$.

Next, we determine λ^+ and ψ_z^+. First, we derive the condition of λ^+ to satisfy inequality 3.25, which is equivalent to

$$
\begin{aligned}
&1 - \sum_{d,z} \frac{U_{dz} \left[\partial_{U_{dz}} L\right]^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \\
=&\sum_{d,z} U_{dz} - \sum_{d,z} \frac{U_{dz} \left[\partial_{U_{dz}} L\right]^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \\
=&\sum_{d,z} \frac{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} U_{dz} - \sum_{d,z} \frac{U_{dz} \left[\partial_{U_{dz}} L\right]^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \\
=&\sum_{d,z} \frac{(\left[\partial_{U_{dz}} L\right]^+ - \left[\partial_{U_{dz}} L\right]^- + \lambda^+) U_{dz}}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \\
=&\sum_{d,z} \frac{\left(\partial_{U_{dz}} L + \lambda^+\right) U_{dz}}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \geq 0.
\end{aligned}
$$

Hence, by satisfying

$$
\forall d, z,\ \lambda^+ \geq -\partial_{U_{dz}} L, \tag{3.29}
$$

inequality 3.25 is satisfied. Likewise, we can ensure

$$
\forall z,\ \psi_z^+ \geq -\partial_{V_{zw}} L \tag{3.30}
$$

which is the inequality condition for satisfying inequality 3.27.

Next, we determine the optimal values of λ^+ and ψ_z^+. Update rules 3.19 and 3.20 can be written as the following gradient descent manner:

$$
\begin{aligned}
U'_{dz} \leftarrow U_{dz} &\frac{\left[\partial_{U_{dz}} L\right]^- + \lambda^-}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} \\
&= U_{dz} - \frac{U_{dz}}{\left[\partial_{U_{dz}} L\right]^+ + \lambda^+} (\partial_{U_{dz}} L + \lambda) \\
&= U_{dz} - \eta \cdot (\partial_{U_{dz}} L + \lambda),
\end{aligned} \tag{3.31}
$$

$$
\begin{aligned}
V'_{zw} &\leftarrow V_{zw} \frac{[\partial_{V_{zw}} L]^- + \psi_z^-}{[\partial_{V_{zw}} L]^+ + \psi_z^+} \\
&= V_{zw} - \frac{V_{zw}}{[\partial_{V_{zw}} L]^+ + \psi_z^+} (\partial_{V_{zw}} L + \psi_z) \\
&= V_{zw} - \delta \cdot (\partial_{V_{zw}} L + \psi_z), \qquad (3.32)
\end{aligned}
$$

where η and δ indicate the step size parameters for gradient descent. Therefore, the smaller λ^+ and ψ_z^+ leads to the larger step size. Thus, we can determine λ^+ and ψ_z^+ as follows:

$$\lambda^+ = \max\left(\max\left(-\partial_{U_{\cdot\cdot}} L\right), 0\right), \qquad (3.33)$$

$$\psi_z^+ = \max\left(\max\left(-\partial_{V_{z\cdot}} L\right), 0\right) \qquad (3.34)$$

which are the smallest value of λ^+ satisfying inequalities 3.25 and 3.26, and ψ_z^+ satisfying inequalities 3.27 and 3.28. Finally, by setting the derivatives of L to be zero, we can achieve:

$$[\partial_{U_{dz}} L]^+ = (UVV^\top)_{dz}, \qquad (3.35)$$

$$[\partial_{U_{dz}} L]^- = (XV^\top)_{dz}, \qquad (3.36)$$

$$[\partial_{V_{zw}} L]^+ = (U^\top UV)_{zw}, \qquad (3.37)$$

$$[\partial_{V_{zw}} L]^- = (U^\top X)_{zw}. \qquad (3.38)$$

3.3.2 Theoretical supports

The validity of update rules of U and V are supported by the following theorems.

Theorem 1 (Satisfaction of equality condition for U)**.** *Under the update rule of U, $\sum_{d,z} U_{dz} = 1$.*

Proof. Under the update rule, λ^+ and λ^- satisfy inequalities 3.25 and 3.26. Therefore, U_{dz} satisfies $\sum_{d,z} U_{dz} = 1$. □ □

Theorem 2 (Satisfaction of equality condition for V)**.** *Under the update rule of V, $\sum_w V_{zw} = 1$.*

Proof. Under the update rule, ψ_z^+ and ψ_z^- satisfy inequalities 3.27 and 3.28. Therefore, V_{zw} satisfies $\sum_w V_{zw} = 1$. □ □

Theorem 3 (Satisfaction of non-negative constraint for U)**.** *Under the update rule of U, $U_{dz} \geq 0$.*

Proof. When λ^+ and λ^- satisfy inequalities 3.25 and 3.26, U_{dz}, $[\partial_{U_{dz}} L]^+$,$[\partial_{U_{dz}} L]^-$, λ^+ and λ^- are all non negative. Thus, updated U_{dz} satisfies $U_{dz} \geq 0$. □ □

Theorem 4 (Satisfaction of non-negative constraint for V)**.** *Under the update rule of V, $V_{zw} \geq 0$.*

Proof. When ψ_z^+ and ψ_z^- satisfy inequalities 3.27 and 3.28, V_{zw}, $[\partial_{V_{zw}} L]^+$,$[\partial_{V_{zw}} L]^-$, ψ_z^+ and ψ_z^- are all non negative. Thus, the updated V_{zw} satisfies $V_{zw} \geq 0$. □ □

Theorem 5 (Convergence)**.** *When U_{dz} and V_{zw} converges, then the final solutions satisfy the KKT optimality conditions.*

(Proof in Appendix 6.1.)

Theorem 6 (Non increasing of loss function)**.** *Loss function L is non-increasing under the iterative update rules.*

(Proof in Appendix 6.2.)

Therefore, under our proposed iterative optimization scheme, the value of loss function never increases while matrices U and V satisfy the probabilistic constraints.

3.3.3 Computational Complexity

In this section, we analyze the computational complexity of the proposed optimization algorithm. The update rules in our algorithm have the following complexities:

For update rule for U.

- Calculate $[\partial_{U_{dz}} L]^+$ requires $O(NMK)$.
- Calculate $[\partial_{U_{dz}} L]^-$ requires $O(NMK)$.
- Calculate λ requires $O(NK)$.

For update rule for V.

- Calculate $[\partial_{V_{zw}} L]^+$ requires $O(NMK)$.
- Calculate $[\partial_{V_{zw}} L]^-$ requires $O(NMK)$.
- Calculate ψ requires $O(MK)$.

In summary, update rule of our method requires the following time-complexity, where *iter* is the number of outer iteration:

$$O(\textit{iter}NMK) \tag{3.39}$$

which is equal to the time complexity of update rule in standard NMF. Consequently, we can say that our optimization algorithm is efficient enough.

3.4 Extensions of PMF

In this section, we give a generalized forms and extensions of PMF. PMF generally represent as following optimization problem:

$$\begin{gathered} U, V = \arg\min_{U,V} \ \mathcal{D}(X||UV) + \mathcal{R}_1(U) + \mathcal{R}_2(V) \\ subject\ to \ \text{Probability Constraints.} \end{gathered} \tag{3.40}$$

,where $\mathcal{D}(X||UV)$ is the loss measure between input matrix X and the product of output matrices UV, $\mathcal{R}_1(U)$ and $\mathcal{R}_2(V)$ are regularization term for U and V respectively, probability constraints represents the constraints that determines the types of normalization.

3.4.1 Loss Measures

For the loss measure, we can consider any differentiable loss measures that represent the difference between the input matrix and the product of output matrices.

In this study, we consider two types of loss measures: Frobenius norm loss and generalized KL-divergence loss. The definitions are:

$$\mathcal{D}(X||UV) = \begin{cases} \mathcal{D}_{Fro}(X||UV) = \|X - UV\|_F^2 \\ \mathcal{D}_{KL}(X||UV) = \sum_{d,w} X_{dw} \log \frac{X_{dw}}{U_{d\cdot} V_{\cdot w}} - X_{dw} + U_{d\cdot} V_{\cdot w} \end{cases} \tag{3.41}$$

respectively, and their derivatives are:

$$\begin{aligned}
&[\partial_{U_{dz}} \mathcal{D}_{KL}(X||UV)]^- = \sum_w \frac{X_{dw}}{(UV)_{dw}} V_{zw}, \; [\partial_{U_{dz}} \mathcal{D}_{KL}(X||UV)]^+ = \sum_w V_{zw}, \\
&[\partial_{V_{zw}} \mathcal{D}_{KL}(X||UV)]^- = \sum_d \frac{X_{dw}}{(UV)_{dw}} U_{dz}, \; [\partial_{V_{zw}} \mathcal{D}_{KL}(X||UV)]^+ = \sum_d U_{dz}, \\
&[\partial_{U_{dz}} \mathcal{D}_{Fro}(X||UV)]^- = (XV^\top)_{dz}, \qquad [\partial_{U_{dz}} \mathcal{D}_{Fro}(X||UV)]^+ = (UVV^\top)_{dz}, \\
&[\partial_{V_{zw}} \mathcal{D}_{Fro}(X||UV)]^- = (U^\top X)_{zw}, \qquad [\partial_{V_{zw}} \mathcal{D}_{Fro}(X||UV)]^+ = (U^\top UV)_{zw}.
\end{aligned}$$

3.4.2 Dirichlet Regularization Term

For regularization terms, we can consider any kinds of regularization terms such as ℓ_2, ℓ_1 norm regularization and manifold regularization. In this study, we propose a new type of regularization terms for output matrices called Dirichlet regularization term. The basic idea is since the output matrices can be represented as a point on probability simplex, by introducing regularization term that represents the negative likelihood of Dirichlet distribution, it enables us to control the tendency of the probability distribution of output matrix. Dirichlet regularization term is formulated as the following equation:

$$\mathcal{R}_{Dir}(U, \alpha, \beta) = -\beta \cdot \sum_{d,z} (\alpha - 1) \log(U_{dz}). \tag{3.42}$$

The regularization term represents the log-likelihood of Dirichlet distribution given a parameter α which control the probability distribution of Dirichlet distribu-

tion, namely,

$$\log Dir(U;\alpha) = \log \prod_d \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \prod_z U_{dz}^{\alpha-1} \propto \sum_{d,z} (\alpha - 1) \log(U_{dz}). \tag{3.43}$$

Note that, based on the property of Dirichlet distribution, by changing the parameter α, we can control the probability distribution of output matrices. If $\alpha > 1$, then the matrices become more smooth. If $\alpha < 1$, then the matrices become more sparse. If $\alpha = 1$, then the regularization term does not contribute to restricting the probability distribution of the matrices. β controls the intensity of regularization.

The derivatives of Dirichlet regularization term are:

$$[\partial_{U_{dz}} \mathcal{R}_{Dir}(U,\alpha)]^- = \beta \frac{\alpha}{U_{dz}},\ [\partial_{U_{dz}} \mathcal{R}_{Dir}(U,\alpha)]^+ = \beta \frac{1}{U_{dz}}. \tag{3.44}$$

3.4.3 Possible Probability Constraints

The probability constraints can represent 4 modes of PMF. Under these modes, the probability interpretation of the input matrix and the product of the output matrices are consistent. Each of them is described in Figures 3.2. In these figures, the probability interpretation of each element in matrices and the probability constraints are described. The red rectangles represent the probability constraint, the sum of the elements in the rectangle is constrained to be 1. Note that, PMF described in section3.2 is PMF mode5. The constraints are as follows:

$$Mode1: \quad \sum_w X_{dw} = 1, \forall d, \quad \sum_z U_{dz} = 1, \forall d, \quad \sum_w V_{zw} = 1, \forall z. \tag{3.45}$$

$$Mode2: \quad \sum_d X_{dw} = 1, \forall w, \quad \sum_d U_{dz} = 1, \forall z, \quad \sum_z V_{zw} = 1, \forall w. \tag{3.46}$$

$$Mode3: \quad \sum_{d,w} X_{dw} = 1, \quad \sum_{d,z} U_{dz} = 1, \quad \sum_w V_{zw} = 1, \forall z. \tag{3.47}$$

$$Mode4: \quad \sum_{d,w} X_{dw} = 1, \quad \sum_d U_{dz} = 1, \forall z \quad \sum_{z,w} V_{zw} = 1. \tag{3.48}$$

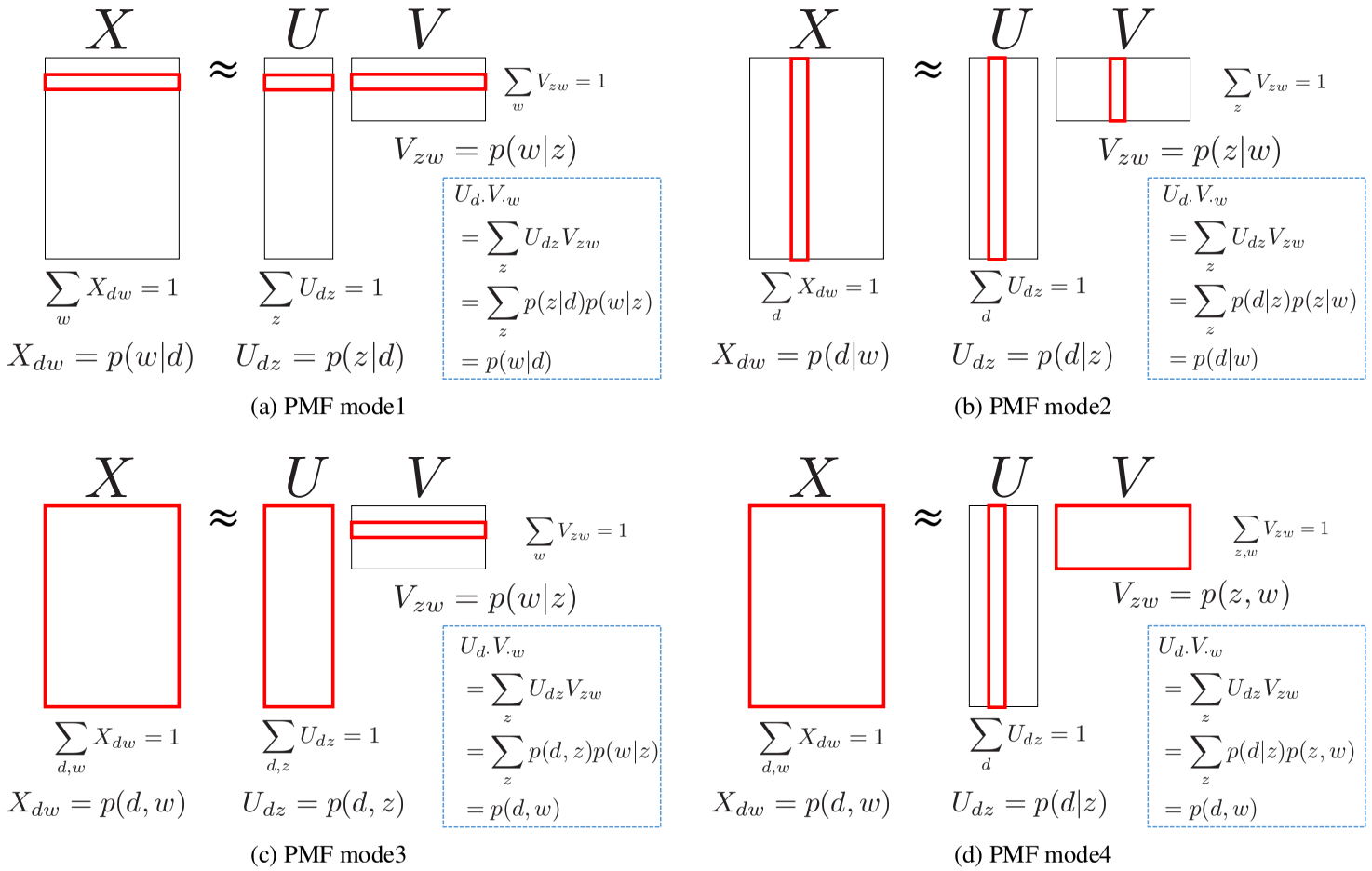


Figure 3.2: **Variations of probability matrix factorization.** There are 4 modes of constraints. Each of them represents different probabilistic interpretations. Under the variations, the probabilistic interpretation of elements in the input matrix and multiplication of the output matrices is consistent.

For optimizing the loss functions for each mode, we can use the optimization algorithm described in section3.3 by setting the Lagrange i.e. λ or ψ_z in an appropriate way. For updating the matrix of constraint that the sum of all elements is 1, we can use the updating rule for U in section3.3. For entire constraint, such as $\sum_{ij} U_{ij} = 1$, we can use the updating rule for U in section3.3. For row-wise constraint, such as $\sum_j U_{ij} = 1$, we can use the updating rule for V in section3.3. For column-wise constraint, such as $\sum_i U_{ij} = 1$, we can use the updating rule for V in section3.3 by setting the index z of ψ_z are ranged to column-wise. In summary, the updating rules for the matrices under constraints are follows:

For entirely constrained matrix.

$$S'_{ij} \leftarrow S_{ij} \frac{\left[\partial_{S_{ij}} L\right]^- + \zeta^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta^+}, \tag{3.49}$$

$$\zeta^- = \frac{1 - \sum_{i,j} \frac{S_{ij}\left[\partial_{S_{ij}} L\right]^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta^+}}{\sum_{i,j} \frac{S_{ij}}{\left[\partial_{S_{ij}} L\right]^+ + \zeta^+}}, \tag{3.50}$$

$$\zeta^+ = \max\left(\max\left(-\partial_{S_{:,:}} L\right), 0\right), \tag{3.51}$$

For row-wisely constrained matrix.

$$S'_{ij} \leftarrow S_{ij} \frac{\left[\partial_{S_{ij}} L\right]^- + \zeta_i^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_i^+}, \tag{3.52}$$

$$\zeta_i^- = \frac{1 - \sum_{j} \frac{S_{ij}\left[\partial_{S_{ij}} L\right]^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_i^+}}{\sum_{j} \frac{S_{ij}}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_i^+}}, \tag{3.53}$$

$$\zeta_i^+ = \max\left(\max\left(-\partial_{S_{i,:}} L\right), 0\right), \tag{3.54}$$

For column-wisely constrained matrix.

$$S'_{ij} \leftarrow S_{ij} \frac{\left[\partial_{S_{ij}} L\right]^- + \zeta_j^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_j^+}, \tag{3.55}$$

$$\zeta_j^- = \frac{1 - \sum_{i} \frac{S_{ij}\left[\partial_{S_{ij}} L\right]^-}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_j^+}}{\sum_{i} \frac{S_{ij}}{\left[\partial_{S_{ij}} L\right]^+ + \zeta_j^+}}, \tag{3.56}$$

$$\zeta_j^+ = \max\left(\max\left(-\partial_{S_{:,j}} L\right), 0\right). \tag{3.57}$$

3.5 Theoretical Analyses for Loss Functions

In this section, we theoretical analyse the loss function of PMF. We revealed that the loss function of generalized KL-divergence is equivalent to probabilistic latent semantic analysis [29], and the KL-divergence based proposed loss function with Dirichlet regularization is equivalent to a loss function of latent Dirichlet allocation with MAP estimation [30].

3.5.1 Relationships between pLSA and PMF_{mode3}

In this sebsection, we analyse the likelihood function of pLSA and the loss function of PMF_{mode3} on generalized KL-divergence. Consequently, we found that the minimization of loss function of PMF_{mode3} is equivalent to maximizing the log-likelihood function of pLSA. In the following section, we summarize generative processes of pLSA and PMF_{mode3}, and we analyze the relation ships between them.

Generative process of pLSA

In pLSA model, the probability of dataset D is

$$p(D) = \prod_d \prod_w \sum_z p(w|z)p(z|d)p(d). \tag{3.58}$$

The probability $p(D)$ is equivalently parameterized by

$$p(D) = \prod_d \prod_w \sum_z p(w|z)p(d|z)p(z). \tag{3.59}$$

pLSA estimates the parameters by maximizing the log-likelihood of the generative probability of dataset $p(D)$. The log-likelihood of $p(D)$ is

$$\log\ p(D) = \sum_{d,w} \log\ \sum_z p(w|z)p(z|d)p(d) = \sum_{d,w} \log\ \sum_z p(w|z)p(d|z)p(z) \tag{3.60}$$

Theoretical analysis of loss function for PMF_{mode3}

In this section, we give a theoretical analysis for loss function of PMF_{mode3} and relationships between $pLSA$.

Let $n_{dw} = N \cdot X_{dw} = N \cdot p(w, d)$, which is the number of a word w in a document d, the generalized KL-divergence loss term for input matrix and output matrices is

$$
\begin{aligned}
N \cdot \mathcal{D}_{KL}(X||UV) &= N \sum_{d,w} X_{dw} \log \frac{X_{dw}}{U_{d\cdot} V_{\cdot w}} - X_{dw} + U_{d\cdot} V_{\cdot w} \\
&= N \sum_{d,w} X_{dw} \log X_{dw} \\
&\quad - N \sum_{d,w} X_{dw} \log U_{d\cdot} V_{\cdot w} \\
&\quad - N \sum_{d,w} X_{dw} + N \sum_{d,w} U_{d\cdot} V_{\cdot w} \\
&= -N \sum_{d,w} X_{dw} \log U_{d\cdot} V_{\cdot w} \\
&\quad + N \sum_{d,w} U_{d\cdot} V_{\cdot w} + \mathcal{C} \\
&= -\sum_{d,w} n_{dw} \log \sum_{z} p(d, z) p(w|z) \\
&\quad + N \sum_{d,w,z} p(z|d) p(z|w) + \mathcal{C} \\
&= -\sum_{d,w} n_{dw} \log \sum_{z} p(d, z) p(w|z) + \mathcal{C}' \\
&= -\sum_{d,w} n_{dw} \log \sum_{z} p(z|d) p(w|z) p(d) + \mathcal{C}' \\
&= -\sum_{d,w} n_{dw} \log \sum_{z} p(d|z) p(w|z) p(z) + \mathcal{C}',
\end{aligned}
$$

which is equivalent to the negative log-likelihood of pLSA with some constant.

In conclusion, minimizing the loss function of PMF_{mode3} is equivalent to max-

imizing the log-likelihood function of pLSA.

3.5.2 Relationships between LDA and PMF_{mode1} with Dirichlet regularization

In this subsection, we analyse the likelihood function of LDA and the loss function of PMF_{mode1} with Dirichlet regularization. Consequently, we found that the minimization of loss function of PMF_{mode1} with Dirichlet regularization corresponds to MAP estimation of LDA with weighted likelihood for prior distributions of output matrices. In the following sections, we summarize generative processes of LDA and PMF_{mode1} with Dirichlet regularization, and we analyse the relationships between them.

Generative process of LDA

Generative process of LDA is as follows:

- For each topic k in K
  - $V_k \sim Dirichlet(V_k|\beta)$
- For each document d in dataset D
  - $U_d \sim Dirichlet(U_d|\alpha)$
  - For each word w_{dn} in a document N_d:
    - $z_{dn} \sim Multinomial(z_{dn}|U_d)$
    - $w_{dn} \sim Multinomial(w|V_{z_{dn}})$.

In the original paper [30], the parameters V_k and U_d correspond to θ and ϕ respectively. Graphical model of LDA is fig. 3.3a. The probability of dataset D is

$$p(D|\alpha,\beta) = \prod_{d}^{D}\prod_{n}^{N_d}\iint\sum_{z} p(w_{dn}|z_{dn},V_{z_{dn}})\; p(z_{dn}|U_d)\; p(U_d|\alpha)\; p(V_k|\beta)\; dUdV \tag{3.61}$$

The posterior probability of parameters is

$$p(U, V|D, \alpha, \beta) = \frac{p(U, V, D|\alpha, \beta)}{p(D|\alpha, \beta)}, \tag{3.62}$$

where

$$p(U, V, D|\alpha, \beta) = \prod_{d}^{D} \prod_{n}^{N_d} \sum_{z} p(w_{dn}|z_{dn}, V_{z_{dn}})\ p(z_{dn}|U_d)\ p(U_d|\alpha)\ p(V_k|\beta). \tag{3.63}$$

Considering MAP estimation for U and V, the optimal parameters U_{MAP} and V_{MAP} are determined by maximizing the log-likelihood of posterior probability given dataset D.

$$\begin{aligned}
U_{MAP}, V_{MAP} &= \arg\max_{U,V}\ \log p(U, V|D, \alpha, \beta) \\
&= \arg\max_{U,V}\ \log p(U, V, D|\alpha, \beta) \\
&= \arg\max_{U,V}\ \log \sum_{\mathbf{z}} p(U, V, \mathbf{z}, D|\alpha, \beta) \\
&= \arg\max_{U,V}\ \sum_{d,n,k} n_{dw} \log \sum_{z} p(w_{dn}|z_{dn}, V_{z_{dn}})\ p(z_{dn}|U_d)\ p(U_d|\alpha)\ p(V_z|\beta) \\
&= \arg\max_{U,V}\ \sum_{d,n} n_{dw} \log\ \sum_{z} p(w_{dn}|z_{dn}, V_{z_{dn}})\ p(z_{dn}|U_d) \\
&\qquad + \sum_{d} \log\ p(U_d|\alpha) \\
&\qquad + \sum_{z} \log\ p(V_z|\beta),
\end{aligned}$$

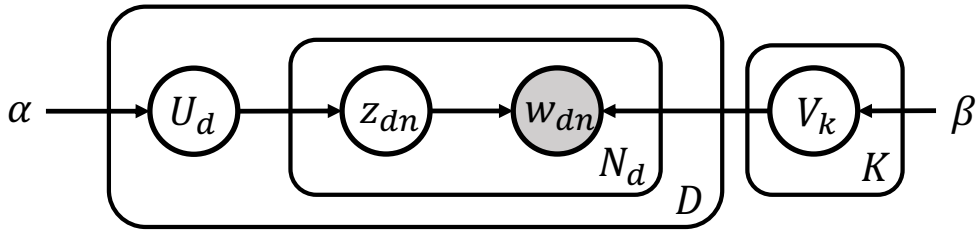
where n_{dw} is a number of appearance of a word w in a document d.

Generative process of PMF_{mode1} with Dirichlet regularization

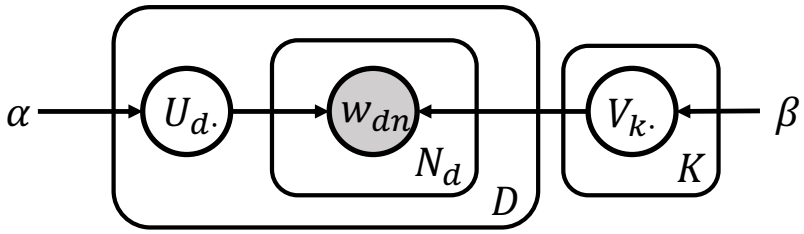
Generative process of PMF_{mode1} with Dirichlet regularization is as follows.

- For each topic k in K
  - $V_{z\cdot} \sim Dirichlet(V_{z\cdot}|\beta)$
- For each document d in dataset D
  - $U_{d\cdot} \sim Dirichlet(U_{d\cdot}|\alpha)$
  - For each word w_{dn} in a document N_d:
    - $w_{dn} \sim Multinomial(w_{dn}|U_{d\cdot}V_{\cdot w})$

Graphical model of PMF_{mode1} with Dirichlet regularization is fig. 3.3b. Note that, marginalizing out z_{dn} in generative process in LDA, the generative process of PMF_{mode1} with Dirichlet regularization and LDA is equivalent.



(a) Graphical model of LDA



(b) Graphical model of PMF_{mode1} with Dirichlet regularization terms

Figure 3.3: **Graphical models of LDA and PMF_{mode1}.** The generative process of PMF_{mode1} is similar to LDA. The parameter z is marginalized out by matrix multiplication.

Theoretical analysis of loss function for PMF_{mode1} with Dirichlet regularization

In this section, we give a theoretical analysis for loss function of PMF_{mode1} with Dirichlet regularization.

Let $n_{dw} = N \cdot X_{dw} = N \cdot p(w|d)$, which is the number of a word w in a document d, the generalized KL-divergence loss term for input matrix and output matrices is

$$\begin{aligned}
N \cdot \mathcal{D}_{KL}(X||UV) &= N \sum_{d,w} X_{dw} \log \frac{X_{dw}}{U_{d\cdot} V_{\cdot w}} - X_{dw} + U_{d\cdot} V_{\cdot w} \\
&= N \sum_{d,w} X_{dw} \log X_{dw} \\
&\quad - N \sum_{d,w} X_{dw} \log U_{d\cdot} V_{\cdot w} \\
&\quad - N \sum_{d,w} X_{dw} + N \sum_{d,w} U_{d\cdot} V_{\cdot w} \\
&= -N \sum_{d,w} X_{dw} \log U_{d\cdot} V_{\cdot w} \\
&\quad + N \sum_{d,w} U_{d\cdot} V_{\cdot w} + \mathcal{C} \\
&= -\sum_{d,w} n_{dw} \log U_{d\cdot} V_{\cdot w} \\
&\quad + N \sum_{d,w,z} p(z|d) p(z|w) + \mathcal{C} \\
&= -\sum_{d,w} n_{dw} \log p(w|U_{d\cdot} V_{\cdot w}) + \mathcal{C}' \\
&= -\sum_{d,w} \sum_{z} n_{dw} \log p(w, z|U_{d\cdot} V_{\cdot w}) + \mathcal{C}' \\
&= -\sum_{d,w} \sum_{z} n_{dw} \log p(w|z, V_{zw})\, p(z|U_{d,\cdot}) + \mathcal{C}'
\end{aligned}$$

So that, the generalized KL-divergence loss term in loss function of PMF_{mode1} is

$$N \cdot \mathcal{D}_{KL}(X||UV) = -\sum_{d,w} \sum_{z} n_{dw} \log p(w|z, V_{zw})\, p(z|U_{d,\cdot}) + \mathcal{C}', \qquad (3.64)$$

which is a negative log-likelihood of word appearance in documents. In other words, minimizing the generalized KL-divergence loss term corresponds to maximizing the log-likelihood of word appearance in documents.

Next, we consider maximization of log-likelihood for posterior probability of LDA with weighting prior probability of output matrices U and V by parameters γ' and δ' respectively. The optimal output matrices $U^*,\ V^*$ which maximizes the log-likelihood of posterior probability is

$$
\begin{aligned}
U^*,\ V^* &= \arg\max_{U,V} \sum_{d,n} n_{dw} \log \sum_{z} p(w_{dn}|z_{dn}, V_{z_{dn}})\, p(z_{dn}|U_d) \\
&\qquad + \gamma' \sum_{d} \log\ p(U_d|\alpha) \\
&\qquad + \delta' \sum_{k} \log\ p(V_k|\beta) \\
&= \arg\max_{U,V} \sum_{d,w} n_{d,w} \log\ p(w, |U_{d,\cdot} V_{\cdot,w}) \\
&\qquad + \gamma' \sum_{d} \log\ p(U_{d\cdot}|\alpha) \\
&\qquad + \delta' \sum_{z} \log\ p(V_{z\cdot}|\beta) \\
&= \arg\max_{U,V} \sum_{d,w} n_{d,w} \log U_{d,\cdot} V_{\cdot,w} \\
&\qquad + \gamma' \sum_{d} \log\ Dir(U_{d\cdot}; \alpha) \\
&\qquad + \delta' \sum_{z} \log\ Dir(V_{z\cdot}; \alpha) \\
&= \arg\min_{U,V} -N \sum_{d,w} X_{dw} \log U_{d,\cdot} V_{\cdot,w} \\
&\qquad - \gamma' \sum_{d} \log\ Dir(U_{d\cdot}; \alpha) \\
&\qquad - \delta' \sum_{z} \log\ Dir(V_{z\cdot}; \alpha) \\
&= \arg\min_{U,V} \mathcal{D}_{KL}(X||UV) + \mathcal{R}_{Dir}(U, \alpha, \gamma) + \mathcal{R}_{Dir}(V, \beta, \delta).
\end{aligned}
$$

Note that, the probability constraints are omitted due to the space limitation. In conclusion, minimizing the loss function of PMF_{mode1} with Dirichlet regularization corresponds to MAP estimation of LDA.

3.6 Experiments

In the experiment, we evaluate our proposed optimization scheme. Specifically, we apply the proposed scheme for topic modeling. We discuss the results of applying the proposed optimization scheme for topic modeling based on PMF. The evaluation criteria are as follows: (1) perplexity, (2) accuracy of topic-based clustering, and (3) convergence speed of the loss function.

3.6.1 Experimental Setting

For the dataset, three benchmark data sets [102] are employed. The first dataset is Reuters dataset. We excluded categories with less than 100 documents, resulting in 8 categories of 7,612 documents. The second dataset is WebKB consisting of 4 categories, including 10,780 documents. The third dataset is 20 newsgroups dataset consisting of 20 categories, including 15,404 documents. The number of words for a dictionary is 5,000. After removing the stop words and stemming, the top of the most frequent words are selected. To evaluate perplexity, we separate the datasets into training documents and test documents in the ratio of 7:3.

As the comparative methods, we employ frobenius norm based NMF [5], generalized KL-divergence based NMF [5] and Latent Dirichlet Allocation (LDA) [30]. As for LDA, we used gensim 3.5.0 implementation of Python 3.6.3.

The experiments were performed on a PC with an Intel Core i7 (3.3 GHz) CPU with 16 GB RAM running macOS. The parameters for Dirichlet regularization are determined by Optuna [103].

3.6.2 Perplexity Evaluation

We examined the quality of topic modeling by various methods. As an evaluation metric, we employed perplexity. Models are trained by using training documents, and by fitting topics for test documents, we calculate the perplexity in test documents. The definition of perplexity is follows:

Definition 3.6.1 (Perplexity)**.**

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d\in D_{test}}\sum_{w}\log p(w|d)}{\sum_{d\in D_{test}} N_d}\right\}, \tag{3.65}$$

where D_{test} is the test dataset, N_d is the number of words in a document d, $p(w|d)$ is prediction probability for word appearance in test dataset by the model.

Perplexity measures the log-likelihood of prediction probability for the words in the test dataset. The lower perplexity indicates better generalization performance.

How to Calculate Perplexity for Test Data

To calculate the perplexity in the test data, it is necessary to calculate the topic distribution in each test document while fixing the word distribution on each topic learned from the training data. The optimization problem to realize that is described as follows:

$$U^{test} = \arg\min_{U}\ \mathcal{D}(X^{test}||UV^{train}) + \mathcal{R}(U), \tag{3.66}$$
$$subject\ to\ \text{Probability Constraints}.$$

X^{test} is a probability matrix of which elements represent the probability between test documents and words, and V^{train} is a probability matrix learned from training data. To optimize the problem, we iteratively applied updating rules corresponding to the probability constraint for U.

To calculate the perplexity of PMF, we derive the predictive probability for test data $p(w|d)$ for each mode of PMF. As for PMF_{mode1}, the probability $p(w|d) = U^{test}V^{train}_{dw}$ is directly used to make prediction. As for PMF_{mode2}, the probabilistic interpretation of multiplication of the output matrices is $p(d|w)$. So that, by letting $p(w) = \frac{1}{M}$, following the Bayes rule, the prediction probability $p(w|d)$ is calculated as follows:

$$p(w|d) = \frac{p(d|w)p(w)}{\sum_w p(d|w)p(w)} = \frac{(U^{test}\ V^{train})_{dw}}{\sum_{w'} (U^{test}\ V^{train})_{dw}}. \tag{3.67}$$

As for $PMF_{mode3,4}$, the multiplication of the output matrices represents the probability $p(w, d)$. Therefore the prediction probability $p(w|d)$ is calculated as follows:

$$p(w|d) = \frac{p(w, d)}{\sum_{w'} p(w', d)} = \frac{(U^{test}\ V^{train})_{dw}}{\sum_{w'} (U^{test}\ V^{train})_{dw}}. \tag{3.68}$$

Results: Perplexity

Table 3.1 are the perplexity scores for Reuters dataset. These tables show that our PMF under generalized KL-divergence of mode 1 and 3 achieve the highest perplexity than the baselines.

3.6.3 Document Clustering Evaluation

In this section, we describe the accuracy of topic-based document clustering. Clustering was performed, having obtained the topic model learned by training data, we assign for each test document the topic with the highest probability through fitting the test document to the model.

Evaluation Metrics: ACC and NMI

To measure the performance of clustering, we use two popular metrics namely clustering accuracy (ACC) and normalized mutual information (NMI) [6].

Definition 3.6.2 (ACC)**.** *Given a set $\mathbb{S}$ of elements, for each element $n \in \mathbb{S}$, the true label and the cluster label generated by a method are denoted by s_n and r_n, respectively. Then, the ACC is defined as:*

$$ACC = \frac{\sum_{n \in \mathbb{S}} \delta(s_n, map(r_n))}{|\mathbb{S}|}$$

where $|\cdot|$ is the cardinality of a set; $\delta(x, y)$ is a delta function which returns 1 if $x = y$, otherwise 0; and $map(r_n)$ is a mapping function that maps r_n to the equivalent label in the dataset. The best mapping can be found by Kuhn-Munkres algorithm [104].

NMI is normalization of the mutual information (MI) to scale the result between 0 and 1. The definition of MI is as follows:

Definition 3.6.3 (MI)**.** *Given the two sets of clusters $\mathcal{C}$, $\mathcal{C}'$, their mutual information $MI(\mathcal{C}, \mathcal{C}')$ is*

$$MI(\mathcal{C}, \mathcal{C}') = \sum_{c_i \in \mathcal{C}, c'_j \mathcal{C}'} p(c_i, c'_j) \log_2 \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)},$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that arbitrarily selected document from dataset belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c_j)$ is the probability that arbitrarily selected document from dataset belongs to the clusters c_i as well as c'_j at the same time.

Based on that, NMI is defined as follows:

Definition 3.6.4 (NMI)**.** *Given the two sets of clusters $\mathcal{C}$, $\mathcal{C}'$, their normalized mutual information $NMI(\mathcal{C}, \mathcal{C}')$ is*

$$NMI(\mathcal{C}, \mathcal{C}') = \frac{MI(\mathcal{C}, \mathcal{C}')}{max(H(\mathcal{C}), H(\mathcal{C}'))},$$

where $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of $\mathcal{C}$ and $\mathcal{C}'$, respectively.

How to Calculate Cluster Assignment for Test Data

To calculate cluster assignment $p(z|d)$ for test data, we optimize the problem 3.67. As for PMF_{mode1}, the probability $U^{test} = p(z|d)$ is directly used to assign the document to cluster. As for PMF_{mode2}, the probabilistic interpretation of U^{test} is $p(d|z)$. So that, by letting $p(z) = \frac{1}{K}$, following the Bayes rule, the cluster assignment $p(z|d)$ is calculated as follows:

$$p(z|d) = \frac{p(d|z)p(z)}{\sum_{z'} p(d|z')p(z')} = \frac{U^{test}_{dz}}{\sum_{z'} U^{test}_{dz'}}. \tag{3.69}$$

As for $PMF_{mode3,4}$, the probabilistic interpretation of U_{test} is $p(d, z)$. Therefore, the cluster assignment $p(z|d)$ is calculated as follows:

$$p(z|d) = \frac{p(d,z)}{\sum_{z'} p(d,z')} = \frac{U^{test}_{dz}}{\sum_{z'} U^{test}_{dz'}}. \tag{3.70}$$

Results: Clustering Accuracy

The results are summarized in Table 3.5. These results show that PMF under generalized KL-divergence of mode2, 3, and 1 achieved higher NMI and ACC in all datasets. This indicates that PMF detects clusters more accurately than clustering based on ordinary NMF and LDA.

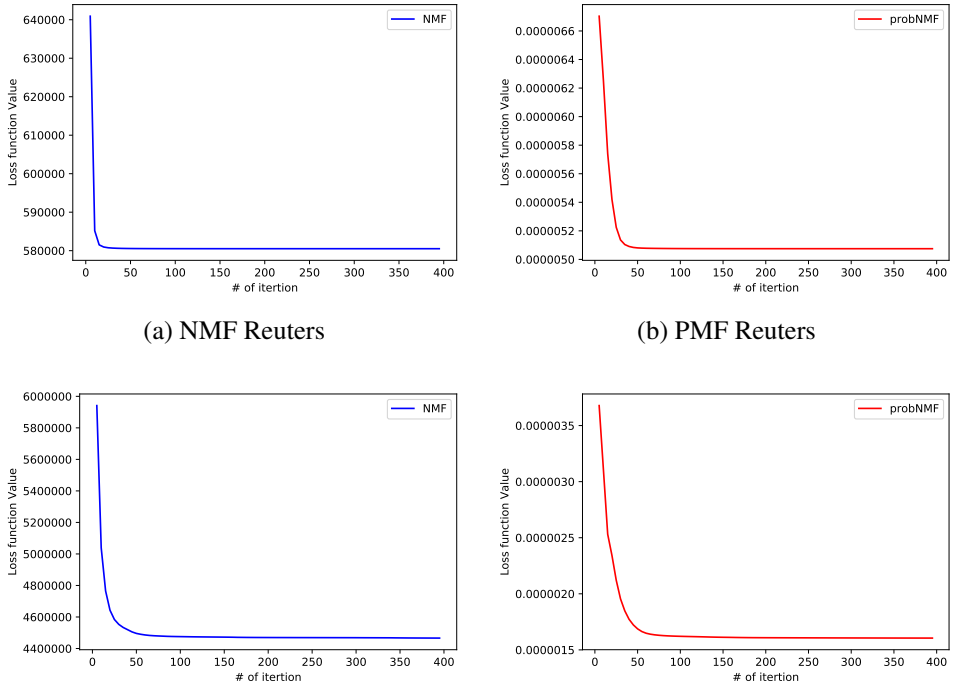
3.6.4 Convergence speed of optimization

In this section, we examine the convergence and learning efficiency of the proposed optimization scheme. In this experiment, we set the number of topics to 8 for Reuters dataset. In Figures 3.4, we plotted the loss function value in each iteration step for the ordinary NMF and the proposed PMF. These figures show that our optimization scheme monotonically decreases the loss function. Moreover, PMF converges as fast as the ordinary NMF. The running times of ordinary NMF on the Reuters (20 newsgroups) dataset are $0.425 \pm 0.015s$ ($3.12 \pm 0.16s$) and PMF are $0.585 \pm 0.030s$ (3.64 ± 0.19). These results imply that our optimization scheme does not incur extra cost and can be practically used as ordinary NMF.

3.7 Conclusion

In this chapter, we have proposed a novel matrix factorization technique called probability matrix factorization (PMF). PMF factorizes the input probability matrix into two probability matrices. We have exploited topic modeling as an example of PMF and derived an iterative optimization algorithm for it. We derived iterative updating rules for the output matrices based on KKT conditions of non-negativity and equality constraints for probability interpretation of the output matrices. For the method,

we have given theoretical supports for the validity of update rules: (1) our optimization scheme monotonically decrease the objective function; (2) the output matrices always meet the probability constraints; (3)the time complexity of the proposed optimization scheme remains the same as the ordinary NMF; and (4) our optimization scheme can apply to many kind of differentiable loss-function. For more general use of PMF, we extended PMF to a more flexible formulation. Specifically, we derived 4 variety of PMFs, which the probability interpretation of the input matrix and the product of the output matrices are consistent, and proposed Dirichlet regularization term, which enables us to control the probability distribution of the output matrices. Moreover, theoretical analysis for the loss-function of PMF revealed that KL-divergence based PMF_{mode3} is equivalent to pLSA, and KL-divergence based PMF_{mode1} with Dirichlet regularization term is equivalent to extension of LDA. The experimental results have shown that, when applying it to topic modeling under PMF, the model optimized by our optimization scheme is more accurate than the ordinary NMF regarding clustering and topic detection without sacrificing efficiency compared with ordinary NMF and LDA.



(a) NMF Reuters

(b) PMF Reuters

(c) NMF 20 Newsgroups

(d) PMF 20 Newsgroups

Figure 3.4: **Loss function value under iterations.** The convergence speed of NMF and PMF is comparable.

Table 3.1: Perplexity on Reuter dataset.

Methods		Number of Topics 5	10	25	50	100
LDA		1041	922	779	674	582
NMF_{Fro}		2466	2175	1721	1432	1188
NMF_{KL}		1420	1351	1376	1248	1006
$\boldsymbol{PMF_{Fro}}$	mode1	1346	1154	995	854	730
	mode2	2.84×10^5	2.08×10^6	4.98×10^6	1.03×10^7	1.28×10^7
	mode3	4952	4679	4447	4330	4237
	mode4	4976	4705	4484	4373	4284
$\boldsymbol{PMF_{KL}}$	mode1	966	**808**	**625**	**505**	**396**
	mode2	3009	2481	1918	1557	1212
	mode3	959	830	697	597	479
	mode4	**951**	830	696	611	519

Table 3.2: Perplexity of 20 Newsgroups dataset.

Methods		Number of Topics 5	10	25	50	100
LDA		1880	1686	1475	1305	1128
NMF_{Fro}		2595	2274	1771	1549	1340
NMF_{KL}		2143	2071	1857	1622	1315
$\boldsymbol{PMF_{Fro}}$	mode1	2274	2112	1859	1656	1408
	mode2	5.24×10^6	2.54×10^7	8.1×10^7	9.8×10^7	8.01×10^7
	mode3	5029	4834	4685	4603	4542
	mode4	5033	4835	4672	4578	4502
$\boldsymbol{PMF_{KL}}$	mode1	**1846**	**1629**	**1366**	**1166**	**959**
	mode2	3508	3023	2487	2135	1785
	mode3	1920	1735	1524	1400	1231
	mode4	1915	1736	1537	1464	1379

Table 3.3: Perplexity of Webkb dataset.

Methods		Number of Topics 5	10	25	50	100
LDA		1542	1456	1346	1253	1139
NMF_{Fro}		2032	1837	1605	1419	1201
NMF_{KL}		2080	2062	2189	2004	1679
$\boldsymbol{PMF_{Fro}}$	mode1	1581	1485	1355	1199	<u>1034</u>
	mode2	6.49×10^6	2.65×10^7	7.35×10^7	8.88×10^7	7.1×10^7
	mode3	5061	4891	4780	4699	4653
	mode4	5083	4895	4736	4641	4564
$\boldsymbol{PMF_{KL}}$	mode1	**1401**	**1259**	**1085**	**940**	**797**
	mode2	3797	3414	2943	2551	2187
	mode3	<u>1438</u>	<u>1352</u>	<u>1254</u>	<u>1167</u>	1058
	mode4	1444	1362	1285	1242	1169

Table 3.4: Perplexity of DBLP dataset.

Methods		Number of Topics 5	10	25	50	100
LDA		<u>678</u>	<u>548</u>	370	254	168
NMF_{Fro}		3.32×10^{19}	3.39×10^{19}	2.31×10^{19}	1.01×10^{19}	4.25×10^{18}
NMF_{KL}		1.78×10^6	2.46×10^8	1.48×10^6	6.0×10^4	423
$\boldsymbol{PMF_{Fro}}$	mode1	854	693	467	322	188
	mode2	7715	1.12×10^4	2.76×10^4	5.55×10^4	9.46×10^4
	mode3	4249	3875	3558	3393	3262
	mode4	4266	3892	3573	3409	3278
$\boldsymbol{PMF_{KL}}$	mode1	**669**	**455**	**225**	**124**	**66**
	mode2	2002	1350	675	378	209
	mode3	1048	698	<u>341</u>	<u>186</u>	<u>101</u>
	mode4	1044	698	344	188	102

Table 3.5: Accuracy of document clustering.

Methods		**ACC score** reuters21578	20news-18828	webkb	dblp	**NMI score** reuters21578	20news-18828	webkb	dblp
LDA		0.489	0.381	0.493	0.388	0.387	0.403	0.197	0.063
NMF_{Fro}		0.493	0.290	0.498	0.424	0.364	0.277	0.212	**0.173**
NMF_{KL}		0.484	0.461	0.649	**0.440**	0.429	0.447	0.342	0.091
$\boldsymbol{PMF_{Fro}}$	mode1	0.338	0.147	0.438	0.314	0.139	0.162	0.155	0.118
	mode2	0.422	0.183	0.432	0.324	0.189	0.153	0.120	0.041
	mode3	0.252	0.096	0.327	0.300	0.058	0.031	0.021	0.009
	mode4	0.257	0.097	0.342	0.299	0.063	0.033	0.026	0.010
$\boldsymbol{PMF_{KL}}$	mode1	0.440	0.483	0.574	0.409	0.414	0.487	0.343	0.088
	mode2	0.563	**0.489**	**0.695**	0.431	**0.519**	**0.494**	**0.391**	0.109
	mode3	**0.578**	0.414	0.663	0.418	0.491	0.406	0.353	0.101
	mode4	0.533	0.398	0.635	0.424	0.478	0.383	0.319	0.103

Chapter 4

Multi-tasking Probability Matrix Factorization

In this chapter, we explore the scheme for collaboratively solve multiple problems via multiple probability matrix factorizations (PMF). Especially, we propose a novel clustering scheme for multi-attributed graphs which consists of three tasks, community detection, attribute clusters and relationships between communities and attributes. The proposed method solves the tasks in a cooperative manner via multiple PMFs and optimizing the multiple loss functions simultaneously. In the following sections, we describe the background of the research, problem definition, model descriptions and the experiments.

4.1 Introduction

Community detection is a task to detect densely connected subgraphs as communities. Nodes in a community tend to share same or similar properties, such phenomenon is called *homophily effect* [18, 105], meaning that nodes having similar properties tend to link together. Because diverse applications are derived from the nature of real communities, community detection is important in graph/network analyses. Examples include node property estimations [47, 53, 106], community-wise information recommendations [107], and semantic reasoning for nodes/edges [108].

Moreover, using the attributes in a graph is advantageous to realize high-quality

community detection as well as to understand the characteristics of communities. Multi-attributed graphs are reasonable models of real-world networks such as social networks, co-author networks, protein-protein interaction networks, etc. In fact, several works have proposed algorithms that employ attribute information (i.e., shared interests or functional behaviors of each community) to detect not only communities but also their semantic meanings [57–59, 63].

However, community detection and extraction of semantics in multi-attributed graphs remain challenging due to difficulties on integrating graph structures and multiple attributes of different types. Community detection and extraction of semantics involve multiple steps. First, useful information from each attribute must be extracted because certain node attributes describe different aspects. Second, all extracted information must be exploited to enhance community detection by effectively integrating heterogeneous information. Notice that the previous works [57–59, 63] do not differentiate multiple attributes, that is, they consider multiple attributes equally. Moreover, real-world graphs are often incomplete and noisy. That is, some edges or nodes may be missing or attribute values may contain incorrect values, leading to inappropriate results.

To overcome these difficulties, we propose a novel clustering scheme based on the following two assumptions:

(1) *Relevant attribute values form clusters by attribute type.* This is based on the observation that an attribute reflects a node's interests in a network. Hence, an attribute tends to be associated to a specific group of values related to an interest. For example, in a co-author network where the nodes correspond to the authors (researchers), each author typically has specific research interests (e.g., AI, data mining, and database). Thus, attributes (e.g., paper title and conference) present biased values according to interests. Consequently, it is possible to identify clusters of attributes values (attribute-value clusters) reflecting a node's interests.

(2) *Communities are strongly correlated with attribute-value clusters.* This is related to the previous assumption. Consider the example above. The nodes in a community share similar interests (e.g., research interests) and consequently, similar attribute-value clusters (e.g., research topics, and conferences). Conversely, if some nodes (researchers) have similar attribute values, they should share similar interests and can be grouped in the same community.

Exploiting the correlation between communities and multiple attributes should improve the quality of community detection as well as attribute-value clustering.

Using the information from different sources (attributes) to alleviate the effect of noise (e.g., missing values and errors), we simultaneously implement community detection and attribute-value clustering.

Based on the aforementioned ideas, we study a novel clustering scheme for multi-attributed graphs, called CAR-clustering. CAR includes **C**ommunity detection, **A**ttribute-value clustering, and deriving **R**elationships between communities and attribute-value clusters for multi-attributed graphs. The image of CAR-clustering is Figure. 4.1. Additionally, we develop a novel clustering algorithm called CARPMF, which employs probability matrix factorizations.

The contributions of this paper are summarized as follows:

- We propose a novel clustering scheme CAR-clustering to address two technical questions. (i) Given a multi-attributed graph, how can community detection and attribute-value clustering be performed for different types of attributes in a cooperative manner? (ii) How should reasonable relationships be determined between communities and attribute-value clusters for each type of attribute?
- We develop a novel algorithm CARPMF, which achieves CAR-clustering. Specifically, a dedicated loss function is designed to perform multiple PMFs simultaneously.
- We conduct experiments using real-world datasets (DBLP computer science bibliography and arXiv physics bibliography). The accuracy of CARPMF with respect to community detection and attribute-value clustering and a comparison to other methods are examined. Relative to comparative methods, CARPMF achieves a better accuracy of up to 19% for community detection and up to 25% for attribute-value clustering. Furthermore, CARPMF detects informative communities and their rich semantic descriptions by correlating multiple types of attribute-value clusters.

In Section 2, we summarize the related works. We provide formal definitions of input graph model and our research objectives in Section 3. We propose our method CARPMF in Section 4. We examine CARPMF in several experiments in Section 5 and conclude the article in Section 6.

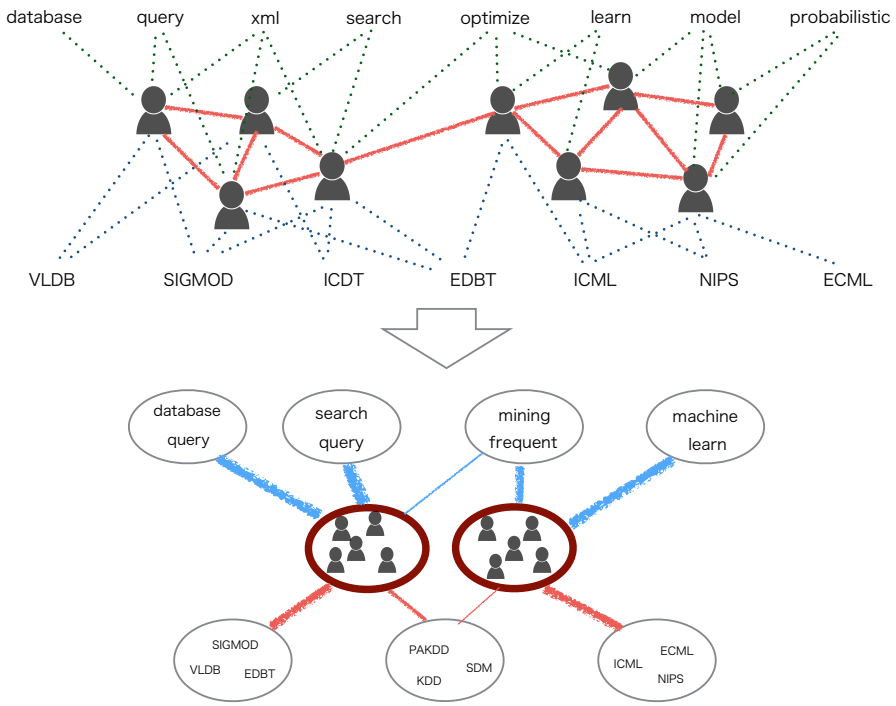


Figure 4.1: **Overview of CAR-clustering.** Given a multi-attributed graph, we try to detect communities, attribute-clusters, and the relationships between them. By detecting them, the communities are characterized by attributed clusters, and the relationships between communities are characterized.

4.2 Problem Statement

In this work, we deal with multi-attributed graphs, where each node is characterized by two or more attributes. Given such a graph, *CAR-clustering* is used to solve the following three sub-problems: community detection, attribute-value clustering, and derivation of relationships between communities and attribute-value clusters, which have been independently studied. Below, we provide the formal definitions which are necessary to define the clustering scheme.

4.2.1 Multi-Attributed Graph

Multi-attributed graph $\mathbb{G}$ is defined by extending weighted graph $\mathbb{G}'$ with several attributed graphs $\mathbb{G}_t$ for attribute $t \in \mathbb{T}$. The following are formal definitions.

Definition 4.2.1 (Weighted graph)**.** *Weighted graph $\mathbb{G}'$ is defined by a triplet, $\langle \mathbb{V}, \mathbb{E}, \mathbb{W} \rangle$, where $\mathbb{V}$ is a set of nodes, $\mathbb{E}(\subseteq \mathbb{V} \times \mathbb{V})$ is a set of edges, and $\mathbb{W} : \mathbb{E} \to \mathbb{R}^+$ is a map of edge weights.* □

Definition 4.2.2 (Attributed graph)**.** *Attributed graph $\mathbb{G}_t = \langle \mathbb{V} \cup \mathbb{X}_t, \mathbb{E}_t, \mathbb{W}_t \rangle$ of attribute $t \in \mathbb{T}$ is a bipartite graph consisting of set $\mathbb{V}$ of nodes, set $\mathbb{X}_t$ of attribute-values, a set of edges $\mathbb{E}_t \subseteq \mathbb{V} \times \mathbb{X}_t$, and $\mathbb{W}_t : \mathbb{E}_t \to \mathbb{R}^+$ is a map of edge weights.* □

Definition 4.2.3 (Multi-attributed graph)**.** *Given weighted graph $\mathbb{G}' = \langle \mathbb{V}, \mathbb{E}, \mathbb{W} \rangle$ and a set of attributed graphs $\{\mathbb{G}_t\}_{t \in \mathbb{T}}$ where $\mathbb{G}_t = \langle \mathbb{V} \cup \mathbb{X}_t, \mathbb{E}_t, \mathbb{W}_t \rangle$, multi-attributed graph*
$\mathbb{G} = \langle \mathbb{G}', \{\mathbb{G}_t\}_{t \in \mathbb{T}} \rangle$ is a union of these graphs. □

4.2.2 CAR-clustering

Given a multi-attributed graph, information can be extracted from different perspectives. In this work, we extract *communities*, *attribute-value clusters*, and the *relationship* between them.

Community. For a multi-attributed graph, a set of nodes with the following properties is regarded as a community. (1) Nodes in a community are densely connected with each other and sparsely connected with other nodes. (2) Nodes in a community tend to share common values in distinct attributes. This study assumes that communities can overlap. That is, each node belongs to more than one community. This assumption is reasonable for real applications. Formally, given the number of communities ℓ, node $n \in \mathbb{V}$ belonging to community $c \in \mathbb{C}$ is described by probability distribution $p(n \mid c)$, where $|\mathbb{C}| = \ell$.

Attribute-value cluster. For attribute $t \in \mathbb{T}$ in a multi-attributed graph, similar or highly correlated attribute values can be grouped into attribute-value clusters. Herein, we assume overlapping clusters. That is, each attribute-value belongs to more than one cluster. Formally, given the number of clusters k_t of attribute $t \in \mathbb{T}$, cluster member $x \in \mathbb{X}_t$ for attribute-value cluster $s_t \in \mathbb{S}_t$ is described by probability distribution $p(x \mid s_t)$, where $|\mathbb{S}_t| = k_t$.

Relationship between a community and an attribute-value cluster. Nodes in a community often share common attribute-value clusters. Detecting such relationship is useful in many applications. Given community $c \in \mathbb{C}$ and attribute-value cluster $s_t \in \mathbb{S}_t$ of attribute $t \in \mathbb{T}$, the probability that c is related to s_t is described as the relationship between c and s_t. In this work, a community may be related to more than one attribute-value cluster. Formally, this is described by probability distribution $p(s_t \mid c)$.

CAR-clustering. CAR-clustering is formally defined by Definition 4.
Definition 4.2.4 (CAR-clustering)**.** *Given a multi-attributed graph $\mathbb{G}$, CAR-clustering is to perform community detection, attribute-value clustering, and detection of the relationship between the communities and the attribute-value clusters simultaneously.*□

Solving these sub-problems simultaneously is more beneficial than evaluating each one independently because, in many cases, communities and attribute-value clusters are mutually correlated. Solving the problems simultaneously exploits this correlation, leading to improved results.

4.3 CARPMF – Algorithm for CAR-clustering

In this section, we propose an NMF (non-negative matrix factorization)-based algorithm, called CARPMF, for CAR-clustering. CARPMF models communities and attribute-value clusters. Additionally, we introduce an auxiliary matrix to maintain the relationship between the communities and the attribute-value clusters. A unified loss function is used to solve the different NMFs in a unified manner. It is assumed that the user gives the number ℓ of communities and the number k_t of clusters for each attribute $t \in \mathbb{T}$.

4.3.1 Matrix representation

We represent a multi-attributed graph by two sorts of matrices: an adjacency matrix $A \in \mathbb{R}^{|\mathbb{V}|\times|\mathbb{V}|}$ and attribute matrices $X^{(t)} \in \mathbb{R}^{|\mathbb{V}|\times|\mathbb{X}_t|}$ for $t \in \mathbb{T}$. An element $A_{u,v}$ of A corresponds to an edge $e_{u,v} = (u, v) \in \mathbb{E}$. $A_{u,v} = \mathbb{W}(e_{u,v}) / \sum_{e_{i,j}\in\mathbb{E}} \mathbb{W}(e_{i,j})$, indicating the joint probability for the presence of edge $e_{u,v}$. Similarly, for $t \in \mathbb{T}$, an element $X^{(t)}_{u,x}$ in $X^{(t)}$ corresponds to an edge $e^{(t)}_{u,x} \in \mathbb{E}_t$. $X^{(t)}_{u,x} = \mathbb{W}_t(e^{(t)}_{u,x}) / \sum_{v,y\in\mathbb{E}_t} \mathbb{W}_t(e^{(t)}_{v,y})$,

indicating the joint probability of the presence of edge $e^{(t)}_{u,x}$.

4.3.2 Loss Function

We achieve CAR-clustering in terms of several NMFs, which correspond to the aforementioned sub-problems. To achieve CAR-clustering, we introduce loss functions for the sub-problems followed by a unified loss function.

Loss function for community detection. In CARPMF, communities $\mathbb{C}$ are denoted by a matrix $U^* \in \mathbb{R}^{|\mathbb{V}| \times \ell}$, where each row and column correspond to a node $u \in \mathbb{V}$ and a community $c \in \mathbb{C}$, respectively. A cell $U^*_{u,c}$ represents joint probability of node u and community c $p(u, c)$. In probability $p(u, v, c)$, u and v are connected through community c, and is represented by $U^*_{u,c}U^*_{v,c}$. Moreover, joint probability $p(u, v)$, or the existence of edge $e_{u,v} \in \mathbb{E}$, is expressed as $\sum_{c\in\mathbb{C}} U^*_{u,c}U^*_{v,c}$. Therefore, when U^* minimizes the following loss function, U^* is the best approximation of the edges in the graph.

$$\begin{aligned} &\arg\min_{U^*\geq 0} \left\| A - U^*(U^*)^\top \right\|_F^2, \\ &subject\ to\ \sum_{i,j} U^*_{ij} = 1, \end{aligned} \tag{4.1}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm.

Loss function for attribute-value clustering. In CARPMF, attribute-value clusters $\mathbb{S}_t$ of attribute $t \in \mathbb{T}$ are represented as a matrix $V^{(t)} \in \mathbb{R}^{|\mathbb{X}_t| \times k_t}$, where each row and column correspond to an attribute $x \in \mathbb{X}_t$ and an attribute cluster $s_t \in \mathbb{S}_t$, respectively. A cell $V^{(t)}_{x,s_t}$ represents probability $p(x \mid s_t)$.

To derive $V^{(t)}$ from $X^{(t)}$, we introduce a matrix $U^{(t)} \in \mathbb{R}^{|\mathbb{V}| \times k_t}$, which denotes the relationships between the nodes and attribute-value clusters with probability $p(u, s_t)$. Using both matrices $U^{(t)}$ and $V^{(t)}$, probability $p(u, x, s_t)$, which is the existence of edge $e^{(t)}_{u,x} \in \mathbb{E}_t$ in terms of attribute-value cluster s_t, is calculated as $U^{(t)}_{u,s_t}V^{(t)}_{x,s_t}$. Moreover, probability $p(u, x)$ is derived as $\sum_{s_t\in\mathbb{S}_t} U^{(t)}_{u,s_t}V^{(t)}_{x,s_t}$. Therefore, when $U^{(t)}$, $V^{(t)}$ minimize loss function, $U^{(t)}$, $V^{(t)}$ represent the best approximation of the edges in the graph.

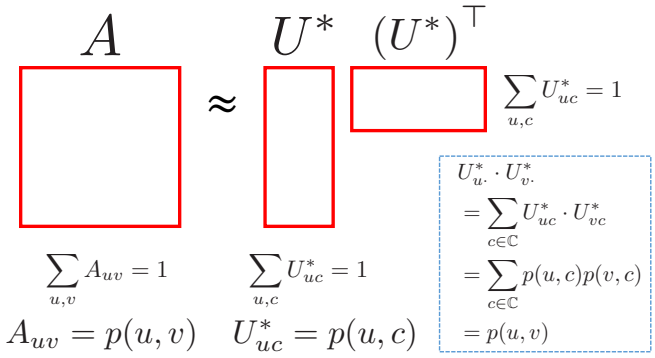
$$\begin{aligned} &\underset{U^{(t)},V^{(t)}\geq 0}{\arg\min} \left\| X^{(t)} - U^{(t)}(V^{(t)})^{\top} \right\|_F^2 \\ &\textit{subject to} \ \sum_{i,j} U_{ij}^{(t)} = 1, \\ &\qquad\qquad \sum_{i} V_{ir}^{(t)} = 1, \ \ \forall 1 \leq r \leq k_t. \end{aligned} \tag{4.2}$$

Loss function for relationship detection. In CARPMF, the relationships between communities and attribute-value clusters of attribute $t \in \mathbb{T}$ are represented as a matrix $R^{(t)} \in \mathbb{R}^{\ell \times k_t}$, where each row and column corresponds to a community $c \in \mathbb{C}$ and an attribute-value cluster $s_t \in \mathbb{S}_t$, respectively. The cell contains the probability $p(s_t \mid c)$. We assume $R^{(t)}$ is a linear transformation that maps U^* into $U^{(t)}$, where U^* and $U^{(t)}$ derived by Equation 4.1 and Equation 4.2, respectively. Moreover, the joint probability $p(u, s_t) = U_{u,s_t}^{(t)}$ can also be calculated as $\sum_c p(u,c)p(s_t \mid c) = \sum_c U_{u,c}^* R_{c,s_t}^{(t)}$ Therefore, when $R^{(t)}$ minimizes the loss function, $R^{(t)}$ represents the relationships between the communities and the attribute-value clusters.

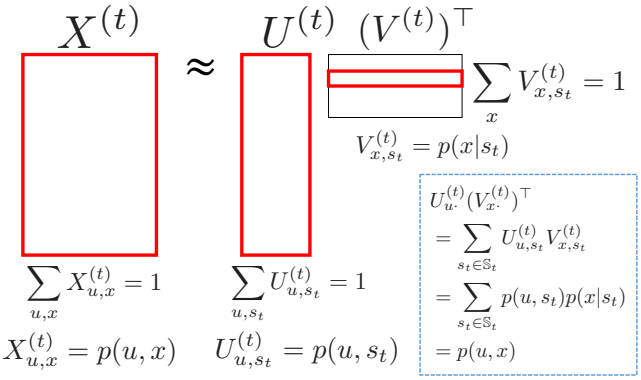
$$\begin{aligned} &\underset{U^{(t)},U^*,R^{(t)}\geq 0}{\arg\min} \left\| U^{(t)} - U^* R^{(t)} \right\|_F^2 \\ &\textit{subject to} \ \sum_{i,j} U_{ij}^* = 1, \ \ \sum_{i,j} U_{ij}^{(t)} = 1, \\ &\qquad\qquad \sum_{j} R_{pj}^{(t)} = 1, \ \ \forall 1 \leq p \leq \ell. \end{aligned} \tag{4.3}$$

Equation 4.3 can be regarded as an NMF that decomposes the matrix of the node-by-attribute value cluster into node-by-community and community-by-attribute value cluster matrices. In other words, Equation 4.3 indicates the effect of the relationship between nodes and attribute-value clusters against communities.

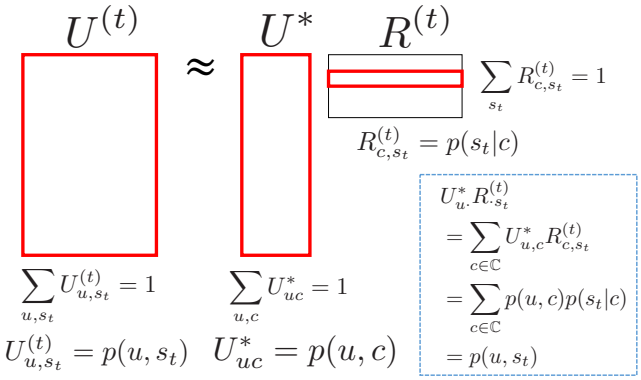
The overview of each task is described in Figures 4.2



(a) Probability matrix factorization for community detection.



(b) Probability matrix factorization for attribute-value clustering.



(c) Probability matrix factorization for relationship detection.

Figure 4.2: **Probability matrix factorizations for CARNMF.** Each of the figures corresponds to the PMF for each task.

Unified loss function. To achieve CAR-clustering, the aforementioned three sub-problems must be solved. In this work, we attempt to solve them simultaneously by introducing a unified loss function, which is expressed as

$$
\begin{aligned}
U^*,\ \left\{U^{(t)},\ V^{(t)},\ R^{(t)}\right\}_{t\in\mathbb{T}} = & \underset{U^*,\ \left\{U^{(t)},\ V^{(t)},\ R^{(t)}\right\}_{t\in\mathbb{T}}}{\arg\min} \left\|A - U^*(U^*)^\top\right\|_F^2 \\
& + \sum_{t\in\mathbb{T}} \left\{ \left\|X^{(t)} - U^{(t)}(V^{(t)})^\top\right\|_F^2 + \lambda_t \left\|U^{(t)} - U^* R^{(t)}\right\|_F^2 \right\}, \qquad (4.4) \\
s.t.\ & \forall 1 \le r \le k_t, \forall 1 \le p \le \ell, \forall t \in \mathbb{T}, \\
& \sum_{i,j} U^*_{ij} = 1, \sum_{i,j} U^{(t)}_{ij} = 1, \sum_i V^{(t)}_{ir} = 1, \sum_j R^{(t)}_{pj} = 1,
\end{aligned}
$$

where λ_t for attribute $t \in \mathbb{T}$ is a user-defined parameter to control the effect of attribute-value clusters for community detection. Higher λ_t yields a stronger effect of the attribute-value clusters in community detection.

4.3.3 Optimization

To optimize the loss function in Eq. 4.4, we derive the updating rules for each of the output matrix, based on the optimization framework for PMF. The updating rules corresponding to the variables are as follows:

$$
\begin{aligned}
U^{*\prime}_{ij} \leftarrow\ & U^*_{ij} \frac{\left[\partial_{U^*_{ij}} L\right]^- + \xi^-}{\left[\partial_{U^*_{ij}} L\right]^+ + \xi^+}, \qquad (4.5) \\
\xi^- = & \frac{1 - \sum_{i,j} \frac{U^*_{ij}\left[\partial_{U^*_{ij}} L\right]^-}{\left[\partial_{U^*_{ij}} L\right]^+ + \xi^+}}{\sum_{i,j} \frac{U^*_{ij}}{\left[\partial_{U^*_{ij}} L\right]^+ + \xi^+}}, \\
\xi^+ = & \max\left(\max\left(-\partial_{U^*_{i,:}} L\right), 0\right),
\end{aligned}
$$

$$
\begin{aligned}
U_{ij}^{(t)\prime} \leftarrow\ & U_{ij}^{(t)} \frac{\left[\partial_{U_{ij}^{(t)}} L\right]^{-} + \eta^{-}}{\left[\partial_{U_{ij}^{(t)}} L\right]^{+} + \eta^{+}}, \\
\eta^{-} &= \frac{1 - \sum_{i,j} \frac{U_{ij}^{(t)} \left[\partial_{U_{ij}^{(t)}} L\right]^{-}}{\left[\partial_{U_{ij}^{(t)}} L\right]^{+} + \eta^{+}}}{\sum_{i,j} \frac{U_{ij}^{(t)}}{\left[\partial_{U_{ij}^{(t)}} L\right]^{+} + \eta^{+}}}, \\
\eta^{+} &= \max\left(\max\left(-\partial_{U_{:,:}^{(t)}} L\right), 0\right),
\end{aligned}
\tag{4.6}
$$

$$
\begin{aligned}
V_{ij}^{(t)\prime} \leftarrow\ & V_{ij}^{(t)} \frac{\left[\partial_{V_{ij}^{(t)}} L\right]^{-} + \phi_i^{-}}{\left[\partial_{V_{ij}^{(t)}} L\right]^{+} + \phi_i^{+}}, \\
\phi_i^{-} &= \frac{1 - \sum_{j} \frac{V_{ij}^{(t)} \left[\partial_{V_{ij}^{(t)}} L\right]^{-}}{\left[\partial_{V_{ij}^{(t)}} L\right]^{+} + \phi_i^{+}}}{\sum_{j} \frac{V_{ij}^{(t)}}{\left[\partial_{V_{ij}^{(t)}} L\right]^{+} + \phi_i^{+}}}, \\
\phi_i^{+} &= \max\left(\max\left(-\partial_{S_{i,\cdot}} L\right), 0\right),
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
R_{ij}^{(t)\prime} \leftarrow\ & R_{ij}^{(t)} \frac{\left[\partial_{R_{ij}^{(t)}} L\right]^{-} + \zeta_i^{-}}{\left[\partial_{R_{ij}^{(t)}} L\right]^{+} + \zeta_i^{+}}, \\
\zeta_i^{-} &= \frac{1 - \sum_{j} \frac{R_{ij}^{(t)} \left[\partial_{R_{ij}^{(t)}} L\right]^{-}}{\left[\partial_{R_{ij}^{(t)}} L\right]^{+} + \zeta_i^{+}}}{\sum_{j} \frac{R_{ij}^{(t)}}{\left[\partial_{R_{ij}^{(t)}} L\right]^{+} + \zeta_i^{+}}}, \\
\zeta_i^{+} &= \max\left(\max\left(-\partial_{R_{i,\cdot}^{(t)}} L\right), 0\right),
\end{aligned}
\tag{4.8}
$$

where

$$[\partial_{U^*} L]^+ = 2U^*(U^*)^\top U^* + \sum_{t\in\mathbb{T}} \lambda_t U^* R^{(t)} (R^{(t)})^\top, \tag{4.9}$$

$$[\partial_{U^*} L]^- = 2A^\top R U^* + \sum_{t\in\mathbb{T}} U^{(t)} (R^{(t)})^\top, \tag{4.10}$$

$$[\partial_{U^{(t)}} L]^+ = U^{(t)}(V^{(t)})^\top V^{(t)} + \lambda_t U^{(t)}, \tag{4.11}$$

$$[\partial_{U^{(t)}} L]^- = X^{(t)} V^{(t)} + \lambda_t U^* R^{(t)}, \tag{4.12}$$

$$[\partial_{V^{(t)}} L]^+ = (V^{(t)})^\top (U^{(t)})^\top U^{(t)}, \tag{4.13}$$

$$[\partial_{V^{(t)}} L]^- = (X^{(t)})^\top U^{(t)}, \tag{4.14}$$

$$[\partial_{R_i} L]^+ = (U^*)^\top U^* R^{(t)}, \tag{4.15}$$

$$[\partial_{R_i} L]^- = (U^*)^\top U^{(t)}. \tag{4.16}$$

The detailed explanations for the derivation of update rules are described in Appendix 6.3,6.4,6.5, 6.6. The aforementioned update rules monotonically decrease the unified loss function (Eq. 4.4).

4.3.4 Complexity Analysis

Here, we analyze the computational complexity of the proposed algorithm. The equations in our algorithm have the following complexities:

- Updating U^* needs $O(|\mathbb{E}|\ell + |\mathbb{V}|\ell^2 \sum_t k_t)$.
- Updating $U^{(t)}$ and $V^{(t)}$ needs $O\left((|\mathbb{V}| + |\mathbb{X}_t|)k_t^2 + |\mathbb{E}_t|k_t\right)$.
- Updating $R^{(t)}$ needs $O\left(|\mathbb{V}|(\ell k_t + \ell^2)\right)$.

In summary, the time complexity of our algorithm is follows, where *iter* is the number of outer iterations (lines 3–16 in our algorithm).

$$O\left(\textit{iter} \sum_t \left(|\mathbb{V}|(\ell^2 k_t + k_t^2) + |\mathbb{X}_t|k_t^2 + |\mathbb{E}|\ell + |\mathbb{E}_t|k_t\right)\right). \tag{4.17}$$

Table 4.1: Selected conferences on four research areas.

DB	DM	ML	IR
SIGMOD, VLDB	KDD, ICDM	NIPS, ICML	SIGIR, ECIR
PODS, EDBT	PKDD, SDM	ECML, UAI	JCDL, ECDL
ICDT	PAKDD	COLT	TREC

4.4 Experimental Evaluations

To demonstrate the applicability and effectiveness of CARPMF, we conducted a set of experiments using real-world datasets. Specifically, the performance of the proposed scheme was compared to simple baseline and the state-of-the-art methods.

The experiments were performed on a PC with an Intel Core i7 (3.3 GHz) CPU with 16 GB RAM running Ubuntu14.04. CARPMF was implemented by Python 2.7.6 with Numpy 1.9.0.

4.4.1 Datasets

We used two datasets: DBLP and arXiv.

- *DBLP*: Digital Bibliography Project[1] is a bibliographic database in the computer science area. DBLP contains publication information, such as authors and conferences. We used a part of the dataset by extracting conferences similar to [109]. We extracted four research areas: data mining, databases, machine learning, and information retrieval, and five major conferences for each area. Consequently, 10,491 papers in 20 conferences (shown in Table 4.1) were selected.

- *arXiv*: arXiv[2] is a repository of electronic preprints in various scientific fields. Similar to above, we chose four research areas: mathematical physics (math-ph), nuclear (nucl-th), astrophysics (astro-ph), and materials (part of cond-mat), and four major journals for each area. Consequently, 12,547 papers in 16 journals (shown in Table 4.2) were selected.

[1] `http://dblp.uni-trier.de/`
[2] `https://arxiv.org`

Table 4.2: Selected journals on four research areas.

math-ph
Communications in Mathematical Physics
Reviews in Mathematical Physics
Letters in Mathematical Physics
Journal of Mathematical Physics
nucl-th
Annual Review of Nuclear and Particle Science
Progress in Particle and Nuclear Physics
Atomic Data and Nuclear Data Tables
Journal of Nuclear Materials
astro-ph
Research in Astronomy and Astrophysics
Annual Review of Astronomy and Astrophysics
New Astronomy Reviews
Space Science Review
cond-mat
Nature Nanotechnology
Nature Materials
Nano Letters
Journal of Materials Science

Multi-attributed graphs were constructed from the datasets as follows: The nodes correspond to the authors. If two authors co-author a paper, we placed a weighted edge between the authors. The weighting denotes the number of co-authored papers. Each author has attributes *term*, *paper*, and *conference/journal*, which are defined below:

- *term*: Each term is regarded as a node. An edge is generated between an author and a term if the author uses the term in the titles of at least one paper. The edge weight denotes the term frequency for each author. As a preprocessing, we applied stop-word elimination and stemming.
- *paper*: Each paper is regarded as a node. An edge is generated if the author publishes the paper. The edge weight is always 1.0 because each paper can only be published once.
- *conference/journal*: Each conference or journal corresponds with a node. An edge is created between an author and a conference/journal if the author publishes at least one paper at the conference/journal. The edge weight is the total number of publications at the conference/journal.

4.4.2 Results of CAR-clustering

Figure 4.3 shows examples of the detected communities and their associated attribute-value clusters in DBLP. The number of communities and the number of term clusters were each 50, whereas the number of conference clusters and the number paper clusters were each 4. The red, blue and gray rectangles correspond to communities, term clusters, and conference clusters, respectively. Each rectangle shows the top contributing nodes in the community/cluster, and the edge weights show the strength of the relationship between the community and the corresponding cluster. We chose famous researchers in different research domains (i.e., *Jiawei Han*, *Michael Stonebraker*, and *Michael I. Jordan*).

Figure 4.3(a) show the community and the correlated attribute-value clusters of *Jiawei Han*, who is a leading researcher in data mining and database areas. The results show that (1) he collaborates with Chinese researchers, (2) he publishes many papers related to data mining conferences (i.e., *KDD, ICDM, SDM, PAKDD* and *PKDD*) and database conferences (i.e., *SIGMOD, VLDB, PODS, EDBT* and *ICDT*), and (3) his researches are highly correlated with topics in data mining, such as *frequent pattern mining* and *high dimensional data*.

Similarly, Figure 4.3(b) shows the result for *Michael Stonebraker*, a renowned database researcher. His community is strongly related to conferences in databases (*SIGMOD, VLDB, PODS, EDBT* and *ICDT*). Topics such as *view maintenance*, and *digital library* are detected. Figure 4.3(c) shows the result for *Michael I. Jordan*, an expert in machine learning research. This community is strongly related to the conferences of machine learning, (*NIPS, ICML, UAI, COLT,* and *ECML*) and the topics like *bayesian inference*, *reinforce learning*, and *support vector machine*.

The proposed scheme is compared to a baseline method as well as the state-of-the-art methods to quantitatively evaluate the performance of community detection and attribute-value clustering. In this experiment, we find the hyper parameters which bring the highest accuracy for each method using grid search. The comparison methods include:

- **NMF** [5]: Baseline approaches that apply NMF for binary relationships between graph components, including author-term (A-T), author-paper (A-P), author-conference (A-C), term-paper (T-P), and term-conference (T-C)[3].

[3]Because NMF assumes the co-occurrences of binary relationships, paper-conference (one-to-one relationship) is excluded.

- **LCTA** [63]: A probabilistic generative model for communities, topics of textual attributes, and their relationships. We set hyper parameter λ to 0.0 for all dataset.

- **SCI** [58]: An NMF based method for detecting communities as well as their semantic descriptions via node's attribute values. We set hyper parameters α and β to 80 for DBLP dataset, and 80 and 0.05 for arXiv dataset, respectively.

- **HINMF** [11]: A model that clusters objects and attributes simultaneously and takes the consensus among the binary NMFs. This work is the most similar to our proposal. We set hyper parameter α to 0.01 for all dataset.

Note that, LCTA and SCI deal with a single concatenated feature of multiple attributes. Therefore, we prepare concatenated feature consisting of *term*, *document* and *conference/journal*, and apply these approaches on the feature. As for CARPMF, we set parameters λ_t to all 1.

To evaluate the qualities of these methods, we compared the *accuracy* [63] of community and attribute-value clustering w.r.t. *paper* and *conference/journal*. We designed a ground truth to measure the *accuracy*. To derive the ground truth, each author is labeled based on research areas of their papers, in other words, if the author mostly published papers for the specific area, the author is labeled as that area. Similarly, the labels for *conference/journal* and *paper* were manually given by referring to the conference categories.

Definition 4.4.1 (Accuracy)**.** *Given a set $\mathbb{S}$ of elements, for each element $n \in \mathbb{S}$, the true label and the cluster label generated by a method are denoted by s_n and r_n, respectively. Then, the accuracy is defined as:*

$$Accuracy = \frac{\sum_{n \in \mathbb{S}} \delta(s_n, map(r_n))}{|\mathbb{S}|}$$

where $|\cdot|$ is the cardinality of a set; $\delta(x, y)$ is a delta function which returns 1 if $x = y$, otherwise 0; and $map(r_n)$ is a mapping function that maps r_n to the equivalent label in the dataset. The best mapping can be found by Kuhn-Munkres algorithm [104]. □

Table 4.4 summarizes the evaluation results. The number of communities and the number of attribute-value clusters for each attribute are each four. Each cell

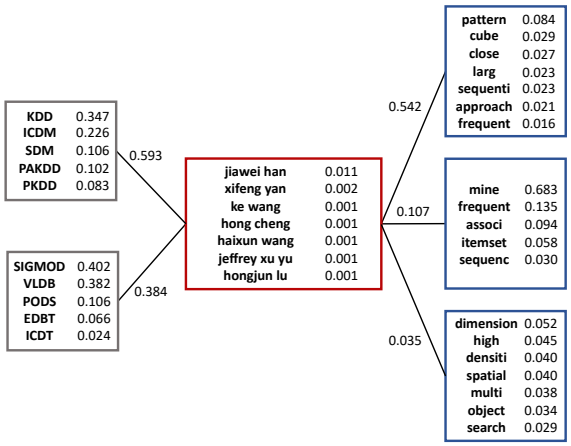
shows the mean and the standard deviation of the accuracies for 20 trials. N/A denotes that the method does not support the category. Values in bold indicate a significant improvement using the Student-t test, where $p < 0.05$.

CARPMF achieved the best performance for community detection (author) and attribute-value clustering (paper and conference/journal) with significant gaps for DBLP dataset (respectively 19%, 25% and 10%) and for arXiv dataset (respectively 9%, 8%) relative to the comparative methods. In particular, CARPMF has an improved clustering quality compared to NMF by taking the relationships between communities and attribute-value clusters into account.

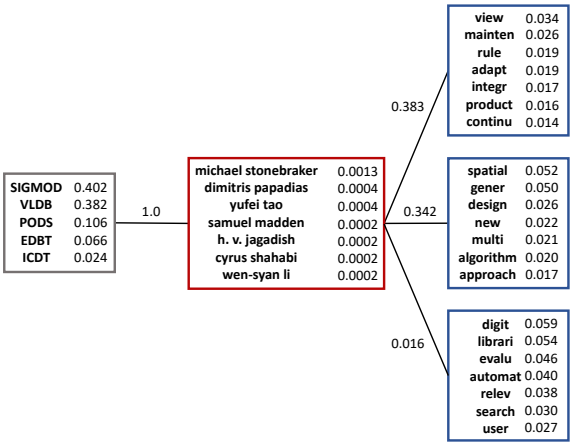
Table 4.3 lists the detected topics from DBLP using CARPMF when the number of topics is set to four. Our method successfully detects the four major research topics. Specifically, Topic 1 containing *retriev, inform, search, queri* and *web* seems to correspond to information retrieval, and Topic 2 containing *mine, pattern, cluster, graph* and *frequent* correspond to data mining. Topic3 contains words "*learn, network, kernel, bayesian, reinforc*", which are typical words of machine learning. Topic4 is a topic of database containing words "*query, databas, optim, xml, manag*", which are popular topics on database researches.

Table 4.3: Detected topics via CARPMF from DBLP.

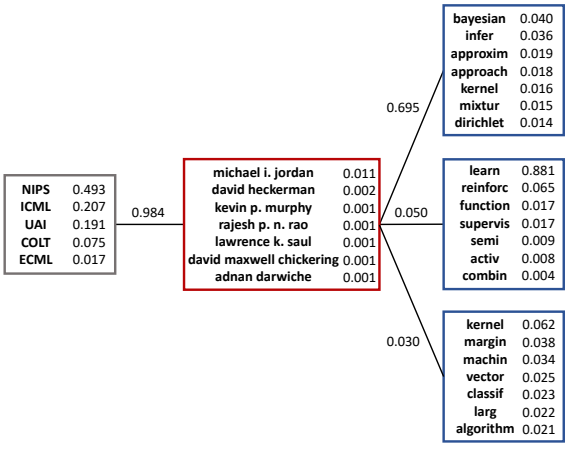
Topic 1 (Information Retrieval)		Topic 2 (Data Mining)		Topic 3 (Machine Learning)		Topic 4 (Database)	
retriev	0.063	mine	0.069	learn	0.049	queri	0.039
trec	0.042	data	0.038	model	0.027	databas	0.038
inform	0.031	cluster	0.033	network	0.015	data	0.035
model	0.025	pattern	0.029	algorithm	0.013	optim	0.014
document	0.024	base	0.022	kernel	0.011	effici	0.013
track	0.023	graph	0.020	bayesian	0.010	xml	0.012
queri	0.023	frequent	0.016	reinforc	0.010	manag	0.012
search	0.021	larg	0.014	infer	0.010	base	0.011
text	0.017	effici	0.014	process	0.009	object	0.011
web	0.017	rule	0.014	decis	0.009	system	0.010



(a) Communities of "jiawei han".



(b) Communities of "michael stonebraker".



(c) Communities of "michael i. jordan".

Figure 4.3: **Example communities with attribute-value clusters.** The red, blue and gray rectangles correspond to communities, term clusters, and conference clusters, respectively.

Table 4.4: Accuracy of community detection and attribute clustering.

	DBLP dataset			**arXiv dataset**		
Methods	Author	Paper	Conference	Author	Paper	Journal
NMF(A-T)	64.02 ± 5.73	N/A	N/A	60.99 ± 0.07	N/A	N/A
NMF(A-P)	43.12 ± 5.17	44.58 ± 5.89	N/A	44.84 ± 5.06	30.94 ± 1.15	N/A
NMF(A-C)	75.35 ± 6.85	N/A	87.60 ± 1.73	75.85 ± 7.29	N/A	73.68 ± 2.33
NMF(T-P)	N/A	50.02 ± 7.93	N/A	N/A	39.80 ± 5.05	N/A
NMF(T-C)	N/A	N/A	69.88 ± 6.68	N/A	N/A	$\mathbf{100.00 \pm 0.0}$
LCTA [63]	48.90 ± 7.57	26.13 ± 4.36	68.50 ± 12.46	46.72 ± 5.72	31.50 ± 1.17	56.87 ± 6.53
SCI [10]	54.78 ± 8.79	22.31 ± 1.48	58.20 ± 7.40	35.42 ± 4.01	29.79 ± 1.11	47.49 ± 6.37
HINMF [11]	68.90 ± 9.08	56.46 ± 3.08	90.10 ± 12.63	74.30 ± 7.99	29.68 ± 0.95	73.12 ± 8.86
CARNMF [13]	86.34 ± 2.39	78.19 ± 9.87	97.20 ± 5.21	77.64 ± 2.88	44.05 ± 3.14	75.00 ± 5.23
CARPMF	$\mathbf{88.56 \pm 1.62}$	$\mathbf{81.64 \pm 4.87}$	$\mathbf{100.00 \pm 0.0}$	$\mathbf{83.42 \pm 0.0}$	$\mathbf{47.84 \pm 0.71}$	87.50 ± 0.0

4.4.3 Insights on Parameters

This section discusses the effect of parameter λ_t for each attribute. The larger the λ_t value, the greater the influence of the attribute-value cluster for $t \in \mathbb{T}$ is on the community. Therefore, optimal parameter settings should result in better results. Figures 4.4 shows the behavior of the accuracy with different values with respect to different attributes. For each evaluation, λ_s ($s \neq t$) of the other attributes were fixed. In most cases, the accuracy shows a convex form and the peak is around 10^0. More importantly, the accuracy is insensitive to the setting, making tuning easier.

4.4.4 How to Determine Parameters

In this section, we discuss about how to determine the user defined parameters (i.e., ℓ, k_t and λ_t). As for the number of communities/clusters (i.e., ℓ, k_t), the larger community/cluster size brings the finer grained communities/clusters (e.g. laboratory, research topic) and the smaller community/cluster size brings the coarse grained communities/clusters (e.g. research society, research area). In the data analysis task, the required granularity of the communities/clusters varies depending on the purpose of the data analysis. Therefore, when applying CARPMF, the number of communities/clusters should be adjusted so as to obtain the target size by repeatedly applying our method. As for λ_t, as we discussed in the previous section, by setting the λ_t to around 1.0, our method achieves highest accuracy. Thus, in the practical use of our method, it is better to set λ_t to 1.0.

4.4.5 Convergence Analysis

In this section, we experimentally provide convergence analysis to optimize the proposed loss function in Equation 4.4. Figures 4.5(a) and (b) show the convergence curve of the loss function for DBLP and arXiv, respectively. In addition, the accuracy of each iteration is plotted. The black line shows the value of the loss function. The red, green, and blue lines show the accuracy of community detection and attribute-value clustering for author, paper, and conference/journal, respectively. As the number of iterations increases, the loss function decreases while the accuracy improves.

4.4.6 Efficiency Analysis

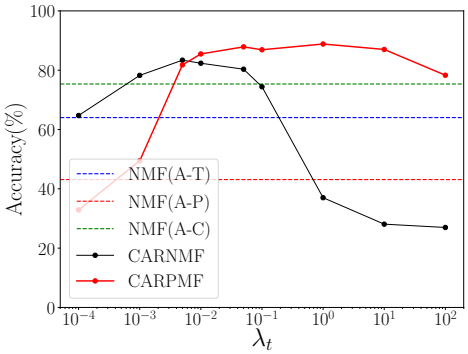
This section analyzes computational efficiency in terms of the numbers of communities and attribute clusters. When the numbers are fixed to four as experiments above, the running times of CARPMF on the DBLP (arXiv) dataset are 15.86 $\pm$ 1.153s (5.163 $\pm$ 0.217s). When changing the numbers of communities and term clusters to 50, while those of paper and conference remain four, the running times increases to 35.05 $\pm$ 0.660s (DBLP) and 70.94 $\pm$ 1.625s (arXiv). These values are still reasonable for various applications.

Moreover, we examine the running time of our method by changing the number of nodes in an input graph. Theoretically, as discussed in Section 4.3.4, the computational complexity is dependent on the number of vertices, that of edges, and that of distinct values of each attribute. As most of real-world graphs are modeled as scale-free networks, edges in a graph are very sparse, therefore, we examine the sensitivity of processing time on the proposed method in terms of the number of nodes. In this experiment, we selected all of the papers on DBLP, and construct the multi-attributed graph as same manner as described in Section 4.4.1. We set the number of communities and clusters are four. Figure 4.6 shows that the time complexity of our method is almost linear to the number of nodes. From the figure, we ensure that the time complexity of our method is linear to the numbers of nodes and edges (as shown on Equation 4.17). Therefore, when the input graph is sparse, our method is highly efficient.

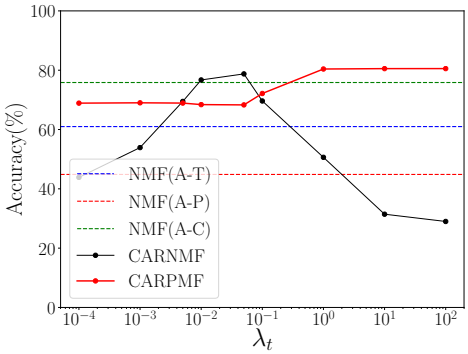
4.5 Conclusion

In this paper we have proposed CAR-clustering, which includes community detection, attribute-value clustering, and extraction of their relationships, for clustering over multi-attributed graphs. We have also proposed a novel algorithm CARPMF based on NMF. CARPMF employs a unified loss function to simultaneously solve different PMFs. This approach is better than the state-of-the-art methods in that it can exploit the correlation between communities and attribute-value clusters for enhancing the quality of the result. Our experiments have demonstrated that CARPMF successfully achieves CAR-clustering. CARPMF has detected reasonable commu-

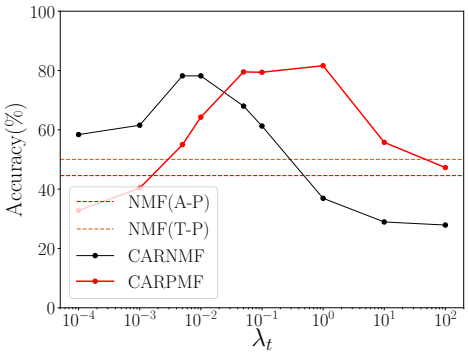
nities with meaningful semantic descriptions via the relationship between communities and attribute-value clusters for real-world datasets. These results are useful for many applications such as node property estimations [47, 53, 106], community-wise information recommendations [107], and semantic reasoning for nodes/edges [108]. Additionally, CARPMF has achieved higher accuracy than comparative methods, including a baseline and the state-of-the-art methods.



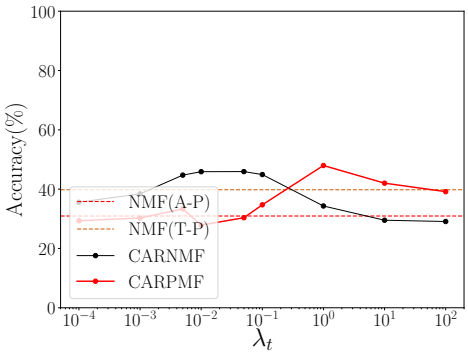
(a) DBLP: Author



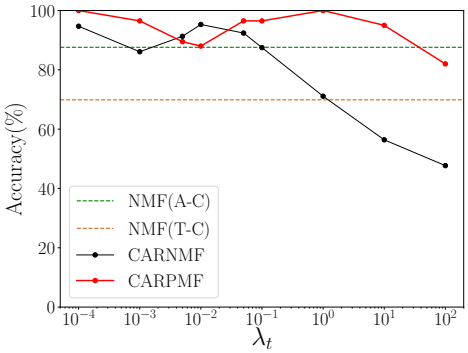
(b) arXiv: Author



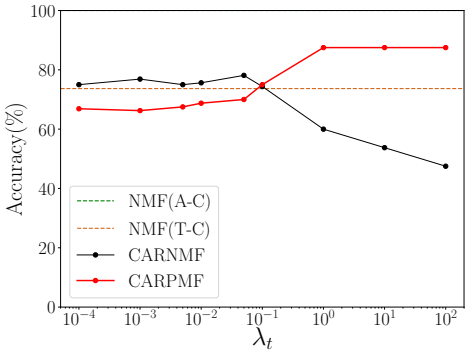
(c) DBLP: Document



(d) arXiv: Document

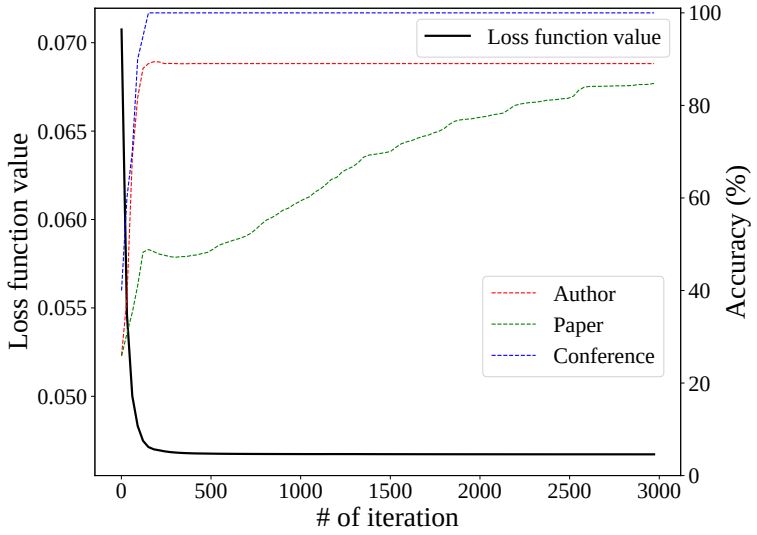


(e) DBLP: Conference

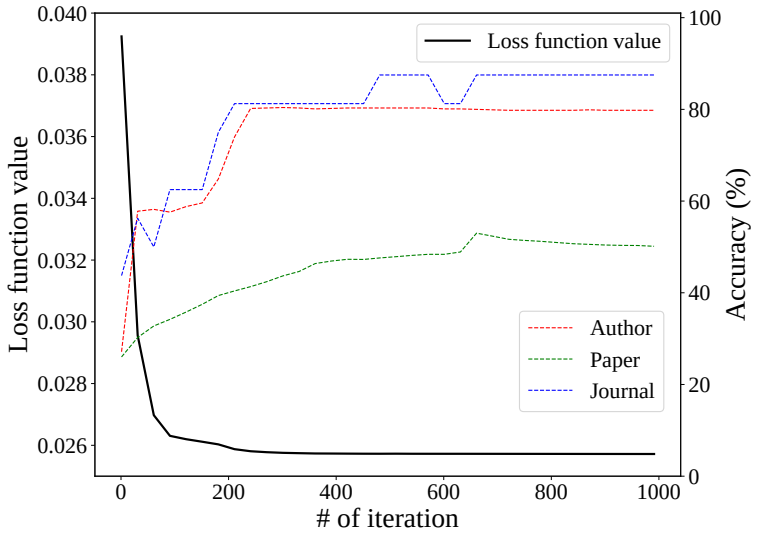


(f) arXiv: Journal

Figure 4.4: **Accuracy for different λ_t values.** CARPMF performs better at all of the parameters are $\lambda_t = 1$.



(a) DBLP



(b) arXiv

Figure 4.5: **Convergence analysis.** Loss function value under iteration and the corresponding accuracy curve. The loss function is decreased monotonically while the accuracy curves are increased.

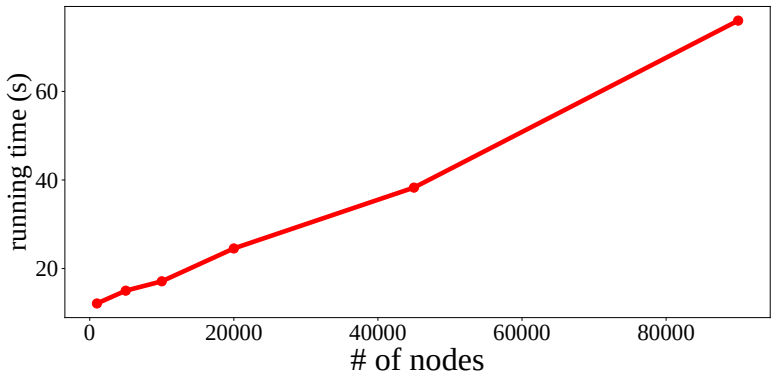


Figure 4.6: **Time complexity of CARPMF w.r.t. the number of input nodes.** CARPMF has almost linear time complexity to the number of nodes in the graph.

Chapter 5

Conclusion and Future Work

This thesis explores non-negative matrix factorization under probability constraints which is named probability matrix factorization (PMF).

Chapter 3 proposed a framework for PMF. PMF is formulated as an optimization problem which consists of loss measure and probability constraints. The loss measure measures a difference between an input probability matrix and a multiplication of output matrices. The probability constraints consist of non-negativity constraint and equality constraint that the sum of the elements in the output matrix is 1. We derived an optimization scheme for PMF that minimizes the loss function while satisfying the constraints, which widely applicable to differentiable loss functions. For more general use of PMF, we extended the PMF to more general loss measure and the variety of probability constraints. To control the probability distribution of the output matrices, we proposed Dirichlet regularization term, which is a negative log-likelihood of the Dirichlet distribution. Experiments using benchmark document datasets showed that PMF outperformed the ordinary NMF and LDA in the viewpoint of the perplexity in the test dataset, and accuracy of document clustering.

Chapter 4 investigates the effectiveness of multi-tasking PMF. In this chapter, a novel clustering scheme for multi-attributed graphs named CAR-clustering is proposed. CAR-clustering includes three tasks, community detection, attribute-value clustering, and relationship detection between the communities and the attribute-value clusters. For CAR-clustering, we proposed CARPMF which consists of multiple PMFs for multiple tasks to mutually complement each other tasks. Experiments using real-world bibliographic databases showed that CARPMF outperforms ordinary NMF and the related works for community detection and clustering.

5.1 Summary of Contributions

Contributions of this thesis are summarized as follows:

- We propose a novel matrix factorization scheme, probability matrix factorization (PMF), which factorized the input probability matrix into two probability matrices (Chapter 3).
- We propose a novel optimization method for optimizing PMF. For the method, we give theoretical supports for the validity of update rules: (1) our optimization scheme monotonically decrease the objective function; (2) the output matrices always meet the probability constraints; (3) the time complexity of the proposed optimization scheme remains the same as the ordinary NMF; (4) our optimization scheme can apply to many kind of differentiable loss-functions (Chapter 3).
- We propose a novel regularization term for the output matrices, Dirichlet regularization term, which enables us to control the probability distribution of the output matrix (Chapter 3).
- The theoretical analysis for loss-function of PMF revealed that PMF strongly relates to probabilistic topic models, namely pLSA [29] and LDA [30] (Chapter 3).
- The experimental results show that PMF is more accurate than the ordinary NMF regarding clustering and perplexity without sacrificing efficiency compared with ordinary NMF (Chapter 3).
- We propose a novel clustering scheme CAR-clustering to address two technical questions. (i) Given a multi-attributed graph, how can community detection and attribute-value clustering be performed for different types of attributes in a cooperative manner? (ii) How should reasonable relationships be determined between communities and attribute-value clusters for each type of attribute? (Chapter 4)
- We develop a novel algorithm CARPMF, which achieves CAR-clustering. Specifically, a dedicated loss function is designed to perform multiple PMFs simultaneously (Chapter 4).

- We conduct experiments using real-world datasets (DBLP computer science bibliography and arXiv physics bibliography). The accuracy of CARPMF with respect to community detection and attribute-value clustering and a comparison to other methods are examined. Relative to comparative methods, CARPMF achieves a better accuracy of up to 19% for community detection and up to 25% for attribute-value clustering. Furthermore, CARPMF detects informative communities and their rich semantic descriptions by correlating multiple types of attribute-value clusters (Chapter 4).

5.2 Future Work

We state our future work for each chapter and that of long-term view.

Future works for the work in Chapter 3 are considered as follows. First, we consider our optimization schemes for NMF under any other kind of distance metrics such as Itakura-Saito (IS) divergence [110], Bregman divergence [111], $\ell_{2,1}$ norms [112], and Wassersterin distance [113]. Second, we are planning to extend PMF to the non-parametric model by considering the prior distribution of the output matrices as the Dirichlet process [114].

Future works for the work in Chapter 4 are considered as follows. First, we are considering to extend the proposed method for chronological analysis over temporal multi-attributed graphs. For example, who immigrates what community, what attribute changes to what cluster, what community changes the relationships between attribute-clusters. Second, we plan to automate the parameter tuning (e.g., the numbers of communities/clusters, λ_t , etc.).

We present the long-term goal. We are considering to extend the PMF to multi-layered PMF, which factorizes the output matrix to more low-rank matrices. The model would explain the hierarchical representation of the latent variables [115, 116] and considered that the model would represent as similar to the deep auto-encoder [117]. The hierarchical feature of latent variables explain global features and local features of input datasets. Because of PMF have probability value, the hidden variables in auto-encoder can be also recognized as probability values. This property may lead to a more interpretable deep auto-encoder model by which the posterior distribution is calculated by the multiplication of the elements.

Chapter 6

Appendix

6.1 Proof of Theorem 5

Proof: At convergence, $U_{ij}^{(\infty)} = U_{ij}^{(t+1)} = U_{ij}^{(t)} = U_{ij}$, $V_{kl}^{(\infty)} = V_{kl}^{(t+1)} = V_{kl}^{(t)} = V_{kl}$, where t denotes the t-th iteration, i.e.,

$$U_{ij} = U_{ij} \frac{\left[\partial_{U_{ij}} L\right]^- + \lambda^-}{\left[\partial_{U_{ij}} L\right]^+ + \lambda^+},$$
$$V_{kl} = V_{kl} \frac{\left[\partial_{V_{kl}} L\right]^- + \psi_k^-}{\left[\partial_{V_{kl}} L\right]^+ + \psi_k^+},$$

which are equivalent to

$$U_{ij} \left(\left[\partial_{U_{ij}} L\right]^+ - \left[\partial_{U_{ij}} L\right]^- + \lambda^+ - \lambda^- \right) = 0,$$
$$V_{kl} \left(\left[\partial_{V_{kl}} L\right]^+ - \left[\partial_{V_{kl}} L\right]^- + \psi_k^+ - \psi_k^- \right) = 0,$$

which are equivalent to Eq. 3.17 and Eq. 3.18, respectively. □

6.2 Proofs of Theorem 6

In this section, we give a proof of Theorem 6. Due to the space limitation, we only prove that the update rule of U does not increase the loss function L. We can

prove for V as the similar fashion. We will follow the similar aproach described in [118] which utilize the property of the auxiliary function. We let $L(U_{ij})$ be the loss function L with U_{ij} as a variable.

Definition 6.2.1. *$G(U_{ij}, U'_{ij})$ is an auxiliary function for $L(U_{ij})$ if the conditions*

$$G(U_{ij}, U'_{ij}) \geq L\left(U_{ij}\right), \quad G(U_{ij}, U_{ij}) = L(U_{ij})$$

are satisfied.

Lemma 6.2.1. *If G is an auxiliary function of F, then F is non-increasing under the update*

$$U'_{ij} = \arg\min_{U_{ij}} G\left(U_{ij}, U_{ij}^{(t)}\right). \tag{6.1}$$

Proof:

$$L\left(U'_{ij}\right) \leq G\left(U'_{ij}, U_{ij}^{(t)}\right) \leq G\left(U_{ij}^{(t)}, U_{ij}^{(t)}\right) = L\left(U_{ij}^{(t)}\right). \square$$

Lemma 6.2.2. *Function*

$$\begin{aligned} G\left(U_{ij}, U_{ij}^{(t)}\right) = L\left(U_{ij}^{(t)}\right) + \partial_{U_{ij}} L\left(U_{ij}^{(t)}\right)\left(U_{ij} - U_{ij}^{(t)}\right) \\ + \frac{\left[\partial_{U_{ij}} L\right]^{+} + \lambda^{+}}{U_{ij}^{(t)}}(U_{ij} - U_{ij}^{(t)})^2 \end{aligned} \tag{6.2}$$

is an auxiliary function for L.

Proof: $G(U_{ij}, U_{ij}) = L(U_{ij})$ is obvious. We need to show that $G(U_{ij}, U_{ij}^{(t)}) \geq L\left(U_{ij}\right)$. To do this, we compare the Taylor series expansion of $L\left(U_{ij}\right)$

$$\begin{aligned} L\left(U_{ij}\right) = L\left(U_{ij}^{(t)}\right) + \partial_{U_{ij}} L\left(U_{ij}^{(t)}\right)\left(U_{ij} - U_{ij}^{(t)}\right) \\ + (\partial_{U_{ij}})^2 L\left(U_{ij}^{(t)}\right)\left(U_{ij} - U_{ij}^{(t)}\right)^2 \end{aligned} \tag{6.3}$$

with Eq. 6.2 to find that $G(U_{ij}, U'_{ij}) \geq L\left(U_{ij}\right)$ is equivalent to

$$\frac{\left[\partial_{U_{ij}} L\right]^{+} + \lambda^{+}}{U_{ij}^{(t)}} \geq (\partial_{U_{ij}})^2 L. \tag{6.4}$$

We have

$$\begin{aligned} \left[\partial_{U_{ij}} L\right]^{+} + \lambda^{+} \geq \left[\partial_{U_{ij}} L\right]^{+} = (UV^{\top}V)_{ij} = \sum_k U_{ik}^{(t)}(V^{\top}V)_{kj} \\ \geq U_{ij}^{(t)}(V^{\top}V)_{jj} = U_{ij}^{(t)}(\partial_{U_{ij}})^2 L. \end{aligned}$$

Therefore, inequality 6.4 holds and $G(U_{ij}, U_{ij}^{(t)}) \geq L\left(U_{ij}\right)$. $\square$

Proof of Theorem1: Replacing $G\left(U_{ij}, U_{ij}^{(t)}\right)$ in equation 6.1 by equation 6.2 results in the update rule:

$$\begin{aligned} U_{ij}' &= U_{ij}^{(t)} - \frac{U_{ij}^{(t)}}{\left[\partial_{U_{ij}} L\right]^{+} + \lambda^{+}} (\partial_{U_{ij}} L + \lambda) \\ &= U_{ij}^{(t)} \frac{\left[\partial_{U_{ij}} L\right]^{-} + \lambda^{-}}{\left[\partial_{U_{ij}} L\right]^{+} + \lambda^{+}}. \end{aligned} \tag{6.5}$$

Since equation 6.2 is an auxiliary function, L is non increasing under this update rule. $\square$

6.3 Fixing $U^{(t)}$, $V^{(t)}$, $R^{(t)}$, optimize L, over U^*

When update U^* with $U^{(t)}$, $V^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned} U^* &= \arg\min_{U^* \geq 0} L(U^*) \\ &= \arg\min_{U^* \geq 0} \left\| A - U^* (U^*)^\top \right\|_F^2 + \sum_{t \in \mathbb{T}} \left\{ \lambda_t \left\| U^{(t)} - U^* R^{(t)} \right\|_F^2 \right\}. \\ & \quad subject\ to\ \sum_{i,j} U_{ij}^* = 1, \end{aligned} \tag{6.6}$$

$L(U^*)$ is equivalent to following equation:

$$\begin{aligned} L(U^*) = & \, tr\left(A^\top A\right) - 2tr\left(A^\top U^* (U^*)^\top\right) \\ & + tr\left((U^*)^\top U^* U^* (U^*)^\top\right) \\ & + \sum_{t \in \mathbb{T}} \lambda_t \left(tr\left((U^{(t)})^\top U^{(t)}\right) - 2tr\left((R^{(t)})^\top (U^*)^\top U^{(t)}\right) \right. \\ & \left. + tr\left((R^{(t)})^\top U^{*\top} U^* R^{(t)}\right) \right). \end{aligned} \tag{6.7}$$

The derivative $\partial_{U^*} L$ is

$$\begin{aligned} \partial_{U^*} L = & -2A^\top R U^* + 2U^* (U^*)^\top U^* \\ & + \sum_{t \in \mathbb{T}} \lambda_t (-U^{(t)} (R^{(t)})^\top + U^* R^{(t)} (R^{(t)})^\top). \end{aligned} \tag{6.8}$$

So that, $[\partial_{U^*} L]^+$ and $[\partial_{U^*} L]^-$ are

$$[\partial_{U^*} L]^+ = 2U^*(U^*)^\top U^* + \sum_{t\in\mathbb{T}} \lambda_t U^* R^{(t)} (R^{(t)})^\top, \tag{6.9}$$

$$[\partial_{U^*} L]^- = 2A^\top R U^* + \sum_{t\in\mathbb{T}} U^{(t)} (R^{(t)})^\top. \tag{6.10}$$

6.4 Fixing U^*, $V^{(t)}$, $R^{(t)}$, optimize L, over $U^{(t)}$

When update $U^{(t)}$ with U^*, $V^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned} U^{(t)} &= \arg\min_{U^{(t)}\geq 0} L(U^{(t)}) \\ &= \arg\min_{U^{(t)}\geq 0} \left\| X^{(t)} - U^{(t)}(V^{(t)})^\top \right\|_F^2 + \lambda_t \left\| U^{(t)} - U^* R^{(t)} \right\|_F^2, \\ &\quad subject\ to\ \sum_{i,j} U_{ij}^{(t)} = 1. \end{aligned} \tag{6.11}$$

$L(U^{(t)})$ is equivalent to following equation:

$$\begin{aligned} L(U^{(t)}) = &tr\left((X^{(t)})^\top X^{(t)}\right) - 2tr\left((X^{(t)})^\top V^{(t)} (U^{(t)})^\top\right) \\ &+ tr\left(U^{(t)}(V^{(t)})^\top U^{(t)} (V^{(t)})^\top\right) \\ &+ \lambda_t tr\left((U^{(t)})^\top U^{(t)}\right) - 2\lambda_t tr\left((R^{(t)})^\top (U^*)^\top U^{(t)}\right) \\ &+ \lambda_t tr\left((R^{(t)})^\top U^{*\top} U^* R^{(t)}\right). \end{aligned} \tag{6.12}$$

The derivative $\partial_{U^{(t)}} L$ is

$$\begin{aligned} \partial_{U^{(t)}} L = &- X^{(t)} V^{(t)} + U^{(t)} (V^{(t)})^\top V^{(t)} \\ &+ \lambda_t (U^{(t)} - U^* R^{(t)}). \end{aligned} \tag{6.13}$$

So that, $[\partial_{U^{(t)}} L]^+$ and $[\partial_{U^{(t)}} L]^-$ are:

$$[\partial_{U^{(t)}} L]^+ = U^{(t)} (V^{(t)})^\top V^{(t)} + \lambda_t U^{(t)}, \tag{6.14}$$

$$[\partial_{U^{(t)}} L]^- = X^{(t)} V^{(t)} + \lambda_t U^* R^{(t)}. \tag{6.15}$$

6.5 Fixing U^*, $U^{(t)}$, $R^{(t)}$, optimize L, over $V^{(t)}$

When update $V^{(t)}$ with U^*, $U^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned} V^{(t)} &= \arg\min_{V^{(t)}\geq 0} L(V^{(t)}) \\ &= \arg\min_{V^{(t)}\geq 0} \left\| X^{(t)} - U^{(t)}(V^{(t)})^\top \right\|_F^2, \\ &\textit{subject to } \sum_i V_{ir}^{(t)} = 1, \ \forall 1 \leq r \leq k_t. \end{aligned}$$

$L(V^{(t)})$ is equivalent to following equation:

$$\begin{aligned} L(V^{(t)}) =& tr\left((X^{(t)})^\top X^{(t)}\right) - 2tr\left((X^{(t)})^\top V^{(t)}(U^{(t)})^\top\right) \\ &+ tr\left(U^{(t)}(V^{(t)})^\top U^{(t)}(V^{(t)})^\top\right). \end{aligned} \tag{6.16}$$

The derivative $\partial_{V^{(t)}} L$ is

$$\partial_{V^{(t)}} L = -2(X^{(t)})^\top U^{(t)} + 2(V^{(t)})^\top (U^{(t)})^\top U^{(t)}. \tag{6.17}$$

So that, $[\partial_{V^{(t)}} L]^+$ and $[\partial_{V^{(t)}} L]^-$ are

$$[\partial_{V^{(t)}} L]^+ = (V^{(t)})^\top (U^{(t)})^\top U^{(t)}, \tag{6.18}$$

$$[\partial_{V^{(t)}} L]^- = (X^{(t)})^\top U^{(t)}. \tag{6.19}$$

6.6 Fixing U^*, $U^{(t)}$, $V^{(t)}$, optimize L, over $R^{(t)}$

When update $R^{(t)}$ with U^*, $U^{(t)}$ and $V^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned} R^{(t)} &= \arg\min_{R^{(t)}\geq 0} L(R^{(t)}) \\ &= \arg\min_{R^{(t)}\geq 0} \left\| U^{(t)} - U^* R^{(t)} \right\|_F^2, \\ &\textit{subject to } \sum_j R_{pj}^{(t)} = 1, \ \forall 1 \leq p \leq \ell. \end{aligned} \tag{6.20}$$

$L(R^{(t)})$ is equivalent to following equation:

$$\begin{aligned} L(R^{(t)}) =& tr\left((U^{(t)})^\top U^{(t)}\right) - 2tr\left((R^{(t)})^\top (U^*)^\top U^{(t)}\right) \\ &+ tr\left((R^{(t)})^\top U^{*\top} U^* R^{(t)}\right). \end{aligned} \tag{6.21}$$

The derivative $\partial_{R^{(t)}} L$ is

$$\partial_{R^{(t)}} L = -(U^*)^\top U^{(t)} + (U^*)^\top U^* R^{(t)}. \tag{6.22}$$

So that, $[\partial_{R_i} L]^+$ and $[\partial_{R_i} L]^-$ are:

$$[\partial_{R_i} L]^+ = (U^*)^\top U^* R^{(t)}, \tag{6.23}$$

$$[\partial_{R_i} L]^- = (U^*)^\top U^{(t)}. \tag{6.24}$$

Bibliography

[1] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[2] Hervé Abdi. The eigen-decomposition: Eigenvalues and eigenvectors. *Encyclopedia of measurement and statistics*, pages 304–308, 2007.

[3] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

[4] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[5] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[6] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada*, pages 267–273, 2003.

[7] Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K Reddy. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1105–1114. International World Wide Web Conferences Steering Committee, 2018.

[8] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[9] Paris Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Independent Component Analysis and Blind Signal Separation*, pages 494–499. Springer, 2004.

[10] Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. Semantic community identification in large attribute networks. In *AAAI*, pages 265–271, 2016.

[11] Jialu Liu and Jiawei Han. Hinmf: A matrix factorization method for clustering in heterogeneous information networks. In *Proceedings of the international joint conference on artificial intelligence workshop*, 2013.

[12] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

[13] Hiroyoshi Ito, Takahiro Komamizu, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Community detection and correlated attribute cluster analysis on multi-attributed graphs. In *Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienna, Austria, March 26, 2018.*, pages 2–9, 2018.

[14] Chris Ding, Tao Li, and Wei Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *AAAI*, volume 42, pages 137–143, 2006.

[15] Chris Ding, Tao Li, and Wei Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

[16] Minnan Luo, Feiping Nie, Xiaojun Chang, Yi Yang, Alexander G Hauptmann, and Qinghua Zheng. Probabilistic non-negative matrix factorization and its robust extensions for topic modeling. In *AAAI*, pages 2308–2314, 2017.

[17] Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[18] Denise B Kandel. Homophily, selection, and socialization in adolescent friendships. *American journal of Sociology*, 84(2):427–436, 1978.

[19] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, 10(5):695–703, 1988.

[20] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[21] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[22] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[23] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, volume 27, page 28. Citeseer, 2002.

[25] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[26] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467. ACM, 2018.

[27] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

[28] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[29] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA*, pages 50–57, 1999.

[30] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[31] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[32] Yee W Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1353–1360, 2007.

[33] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. 2002.

[34] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[35] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006.

[36] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[37] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

[38] Yu Wang, Eugene Agichtein, and Michele Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–131. ACM, 2012.

[39] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[40] Ju-Hong Lee, Sun Park, Chan-Min Ahn, and Daeho Kim. Automatic generic document summarization based on non-negative matrix factorization. *Information Processing & Management*, 45(1):20–34, 2009.

[41] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen. Detect and track latent factors with online nonnegative matrix factorization. In *IJCAI*, volume 7, pages 2689–2694, 2007.

[42] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 745–754. ACM, 2011.

[43] Carmen K Vaca, Amin Mantrach, Alejandro Jaimes, and Marco Saerens. A time-based collective factorization for topic discovery and monitoring in news. In *Proceedings of the 23rd international conference on World wide web*, pages 527–538. ACM, 2014.

[44] Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992–2001, 2013.

[45] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 693–702. ACM, 2012.

[46] Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K Reddy, and Haesun Park. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 567–576. ACM, 2015.

[47] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[48] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[49] Haizheng Zhang, Baojun Qiu, C Lee Giles, Henry C Foley, and John Yen. An lda-based community structure discovery approach for large-scale social networks. *ISI*, 200, 2007.

[50] Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Conference on Learning Theory*, pages 391–423, 2015.

[51] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

[52] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 106–117. SIAM, 2012.

[53] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[54] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114, 2011.

[55] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[56] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[57] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156. IEEE, 2013.

[58] Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. Semantic community identification in large attribute networks. In *AAAI*, pages 265–271, 2016.

[59] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the*

20th conference on Uncertainty in artificial intelligence, pages 487–494. AUAI Press, 2004.

[60] Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-oriented community detection using exhaustive subgroup discovery. *Information Sciences*, 329:965–984, 2016.

[61] Simon Pool, Francesco Bonchi, and Matthijs van Leeuwen. Description-driven community detection. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):28, 2014.

[62] Hongyun Cai, Vincent W Zheng, Fanwei Zhu, Kevin Chen-Chuan Chang, and Zi Huang. From community detection to community profiling. *Proceedings of the VLDB Endowment*, 10(7):817–828, 2017.

[63] Zhijun Yin, Liangliang Cao, Quanquan Gu, and Jiawei Han. Latent community topic analysis: integration of community discovery with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):63, 2012.

[64] Cecile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. Clustering attributed graphs: models, measures and methods. *Network Science*, 3(3):408–444, 2015.

[65] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding and characterizing communities in multidimensional networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 490–494. IEEE, 2011.

[66] Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Mining coherent subgraphs in multi-layer graphs with edge labels. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1258–1266. ACM, 2012.

[67] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 797–806. ACM, 2009.

[68] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[69] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

[70] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

[71] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[72] Xiao Huang, Jundong Li, and Xia Hu. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 731–739, 2017.

[73] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2017.

[74] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[75] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[76] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[77] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 549–553. SIAM, 2006.

[78] Xin Luo, MengChu Zhou, Shuai Li, Zhuhong You, Yunni Xia, and Qingsheng Zhu. A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. *IEEE transactions on neural networks and learning systems*, 27(3):579–592, 2015.

[79] Jiho Yoo and Seungjin Choi. Weighted nonnegative matrix co-tri-factorization for collaborative prediction. In *Asian Conference on Machine Learning*, pages 396–411. Springer, 2009.

[80] Tao Li, Chao Gao, and Jinglin Du. A nmf-based privacy-preserving recommendation algorithm. In *2009 First International Conference on Information Science and Engineering*, pages 754–757. IEEE, 2009.

[81] Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. Speech denoising using nonnegative matrix factorization with priors. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4029–4032. IEEE, 2008.

[82] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V. IEEE, 2006.

[83] Beiming Wang and Mark D Plumbley. Musical audio stream separation by non-negative matrix factorization. In *Proc. DMRN summer conf*, pages 23–24, 2005.

[84] Alexey Ozerov and Cedric Fevotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2009.

[85] William. Kuhn and Albert Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press.

[86] William Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

[87] Tinne Hoff Kjeldsen. A contextualized historical analysis of the kuhn–tucker theorem in nonlinear programming: the impact of world war ii. *Historia mathematica*, 27(4):331–361, 2000.

[88] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

[89] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[90] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602. ACM, 2005.

[91] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[92] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[93] Juntao Liu, Caihua Wu, and Wenyu Liu. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. *Decision Support Systems*, 55(3):838–850, 2013.

[94] Murat Can Cobanoglu, Chang Liu, Feizhuo Hu, Zoltán N Oltvai, and Ivet Bahar. Predicting drug–target interactions using probabilistic matrix factorization. *Journal of chemical information and modeling*, 53(12):3399–3409, 2013.

[95] Mehmet Gönen. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310, 2012.

[96] Hui Fang, Yang Bao, and Jie Zhang. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[97] Juntao Liu, Caihua Wu, Yi Xiong, and Wenyu Liu. List-wise probabilistic matrix factorization for recommendation. *Information Sciences*, 278:434–447, 2014.

[98] Xingyi Ren, Meina Song, E Haihong, and Junde Song. Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation. *Neurocomputing*, 241:38–55, 2017.

[99] Allison JB Chaney, David M Blei, and Tina Eliassi-Rad. A probabilistic model for using social networks in personalized item recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 43–50. ACM, 2015.

[100] Antonio Hernando, Jesús Bobadilla, and Fernando Ortega. A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97:188–202, 2016.

[101] Tuomas Virtanen, A Taylan Cemgil, and Simon Godsill. Bayesian extensions to non-negative matrix factorisation for audio signal modelling. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1825–1828. IEEE, 2008.

[102] Deng Cai and Xiaofei He. Manifold adaptive experimental design for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):707–719, 2012.

[103] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM, 2019.

[104] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[105] Peter V Marsden. Homogeneity in confiding relations. *Social networks*, 10(1):57–76, 1988.

[106] Mario Frank, Andreas P Streich, David Basin, and Joachim M Buhmann. Multi-assignment clustering for boolean data. *Journal of Machine Learning Research*, 13(Feb):459–489, 2012.

[107] Junzo Kamahara, Tomofumi Asakawa, Shinji Shimojo, and Hideo Miyahara. A community-based recommendation system to reveal unexpected interests. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pages 433–438. IEEE, 2005.

[108] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

[109] Jing Gao, Wei Fan, Yizhou Sun, and Jiawei Han. Heterogeneous source consensus learning via decision propagation and negotiation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 339–348. ACM, 2009.

[110] Cédric Févotte and A Taylan Cemgil. Nonnegative matrix factorizations as probabilistic inference in composite models. In *Signal Processing Conference, 2009 17th European*, pages 1913–1917. IEEE, 2009.

[111] Suvrit Sra and Inderjit S Dhillon. Generalized nonnegative matrix approximations with bregman divergences. In *Advances in neural information processing systems*, pages 283–290, 2006.

[112] Deguang Kong, Chris Ding, and Heng Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 673–682. ACM, 2011.

[113] SS Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974.

[114] Michael D Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

[115] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1393–1402. ACM, 2018.

[116] Jinshi Yu, Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Learning the hierarchical parts of objects by deep non-smooth nonnegative matrix factorization. *IEEE Access*, 6:58096–58105, 2018.

[117] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[118] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Reference Papers

Journal Paper

- Hiroyoshi Ito, Takahiro Komamizu, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Detecting Communities and Correlated Attribute Clusters on Multi-Attributed Graphs," *IEICE TRANSACTIONS on Information and Systems, Vol.E102-D, No.4*, pp.810-820, May 2019.

Conference Paper

- Hiroyoshi Ito, Takahiro Komamizu, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Community Detection and Correlated Attribute Cluster Analysis on Multi-Attributed Graphs," *In Proceedings of the 2nd International Workshop on Data Analytics Solutions for Real-Life Applications with 21st EDBT/ICDT Joint Conference (DARLI-AP2018)*, pp.2-9, Vienna, Austria, March 26th - 29th, 2018.
- Hiroyoshi Ito, Toshiyuki Amagasa, "An Optimization Scheme for Non-Negative Matrix Factorization Under Probability Constraints," *In Proceedings of the 27th IEEE International Conference on Big Data and Smart Computing (BigComp2019)*, pp.1-8, Kyoto, Japan, Feburuary 27th - March 2nd, 2019.

Other Papers

Journal Paper

- Hiroyoshi Ito, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Detecting Topic Evolutions in Bibliographic Databases Exploiting Citations," *Information Modelling and Knowledge Bases, IOS Press, Vol.XXVIII*, pp.489-504, January 2017.
- Hiroyoshi Ito, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Detecting Topic Evolutions in Bibliographic Databases Exploiting Citations," *DBSJ Japanese Journal Vol. 14-J, No13*, March 2016 (in Japanese with English Abstract).

Conference Paper

- Hiroyoshi Ito, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Detecting Topic Evolutions in Bibliographic Databases Exploiting Citations," *The 26th International Conference on Information Modelling and Knowledge Bases (EJC2016)*, Tampere, Finland, June 6-10, 2016.
- Hiroyoshi Ito, Takahiro Komamizu, Toshiyuki Amagasa, Hiroyuki Kitagawa, "Network-Word Embedding for Dynamic Text Attributed Networks," *The 6th International Workshop on Semantic Computing for Social Networks and Organization Science with Twelfth IEEE ICSC (SCSN2018)*, pp. 334-339, Laguna Hills, USA, January 31 − February 2, 2018.
- Shintaro Kurimoto, Yasuhiro Hayase, Hiroshi Yonai, Hiroyoshi Ito, Hiroyuki Kitagawa, "Class Name Recommendation based on Graph Embedding of Program Elements," *In proceedings of the 26th Asia-Pacific Software Engineering Conference (APSEC2019)*, pp.498-505, Putrajaya, Malaysia, December 2-5, 2019.