DISSERTATION

ITERATIVE MATRIX COMPLETION AND TOPIC MODELING USING MATRIX AND TENSOR FACTORIZATIONS

Submitted by

Lara Kassab

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2021

Doctoral Committee:

Advisor: Henry Adams

Bailey Fosdick
Michael Kirby
Chris Peterson

ABSTRACT

ITERATIVE MATRIX COMPLETION AND TOPIC MODELING USING MATRIX AND TENSOR FACTORIZATIONS

With the ever-increasing access to data, one of the greatest challenges that remains is how to make sense out of this abundance of information. In this dissertation, we propose three techniques that take into account underlying structure in large-scale data to produce better or more interpretable results for machine learning tasks.

One of the challenges that arise when it comes to analyzing large-scale datasets is missing values in data, which could be challenging to handle without efficient methods. We propose adjusting an iteratively reweighted least squares algorithm for low-rank matrix completion to take into account sparsity-based structure in the missing entries. We also propose an iterative gradient-projection-based implementation of the algorithm, and present numerical experiments showcasing the performance of the algorithm compared to standard algorithms.

Another challenge arises while performing a (semi-)supervised learning task on high-dimensional data. We propose variants of semi-supervised nonnegative matrix factorization models and provide motivation for these models as maximum likelihood estimators. The proposed models simultaneously provide a topic model and a model for classification. We derive training methods using multiplicative updates for each new model, and demonstrate the application of these models to document classification (e.g., 20 Newsgroups dataset).

Lastly, although many datasets can be represented as matrices, datasets also often arise as high-dimensional arrays, known as higher-order tensors. We show that nonnegative CANDECOMP/PARAFAC tensor decomposition successfully detects short-lasting topics in temporal text datasets, including news headlines and COVID-19 related tweets, that other popular methods such as Latent Dirichlet Allocation and Nonnegative Matrix Factorization fail to fully detect.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the people who were a huge part of this journey.

I am incredibly thankful for my advisor, Henry Adams. Henry provided me with unbelievable support for which I will always be grateful. Henry always helped and encouraged me to believe in myself and was constantly present to support me during my lowest of lows as I navigated life and particularly graduate school and life in the U.S. I am very grateful for Henry's tremendous support and mentorship. No amount of words can express my true appreciation for him.

I am extremely thankful for Deanna Needell for believing in me and welcoming me into her research group. I am very grateful to have worked with her and her group at University of California, Los Angeles. I thank Deanna for being an incredible role model for me. She was always very caring and present to support me on a career level and on a personal level. I am very appreciative of her and her mentorship.

I would like to thank Jamie Haddock and Alona Kryshchenko for making my academic journey filled with positive experiences as I navigated life and particularly research and graduate school. I don't think I would have made it this far without their constant support. They made research fun, exciting, and less terrifying for me. I truly cherish our relationship.

I would like to thank Michael Kirby for supporting me during my journey at Colorado State. I am very appreciative of our conversations. I have learned so many things from him. I would also like to thank my other committee members, Bailey Fosdick and Chris Peterson. Thank you for serving on my committee and for supporting me throughout my graduate career.

I would also like to thank all the staff of the Mathematics Department and the International Office at Colorado State, particularly Bryan Elder and Paige Kanatous for all their time and patience. I am also very grateful for Patrick Shipman who was an incredible teaching coordinator and mentor for me. They have all made my graduate experience incredibly beautiful. I also would like to thank all my students. They have made my teaching journey incredibly rewarding.

The research of Chapter 3 stemmed from research that was conducted at the Research Collaboration Workshop for Women in Data Science and Mathematics, July 2019, held at ICERM. I would like to thank all the members of the group: Miju Ahn, Nicole Eikmeier, Jamie Haddock, Alona Kryshchenko, Kathryn Leonard, R. W. M. A. Madushani, Elena Sizikova, Chuntian Wang, and our group lead Deanna Needell.

The research of Chapter 4 initiated from University of California, Los Angeles. I would like to thank all the members of the group: Alona Kryshchenko, Han Lyu, Denali Molitor, Elizaveta Rebrova, and our group lead Deanna Needell. I am also super appreciative of conversations with Jacob Moorman.

My friends and fellow current and past graduate students at Colorado State University, Brendan, Mark, John, Juanita, Naomi, Shannon, Graham, Emily, Joshua, Michael, Casey, Karleigh, Sam, Colin, Patrick, Brittany, Richard, Kyle, Lander, a big thank you. None of this would have been possible without their being part of my everyday life. I am very thankful for our friendship and for the scientific conversations we have had.

Last but not least, thank you to my family, Pierre, Carmen, Elie, Sarah, who supported me throughout my entire journey right from the beginning. I am very grateful for you.

DEDICATION

I would like to dedicate this work to all my loved ones.

TABLE OF CONTENTS

Chapter 1

Introduction

With the ever-increasing access to data, one of the greatest challenges that remains is how to make sense out of this abundance of information. There are a lot of challenges that arise when it comes to analyzing large-scale datasets. One of which is missing values in data, which could be challenging to handle without efficient methods. It is often the case that the missing information is not missing at random, implying that there is an underlying structure in the missing data that, if taken into account, can allow for better recovery. Another challenge arises while performing a (semi-)supervised learning task on extremely high-dimensional data. A common approach is to first apply dimensionality-reduction, and then train the model for the learning task on the low-dimensional representation of the data. One problematic aspect of this two-step approach is that the learned representation of the data may provide a “good" fit, but could suppress data features which are integral to the learning task. For this reason, supervision-aware dimensionality-reduction models have become increasingly important in data analysis and learning tasks. Lastly, although many datasets can be represented as matrices, datasets also often arise as high-dimensional arrays, known as higher-order tensors. If treated as such, nonnegative tensor decompositions extract more *spatio-temporally localized* features than tradition matrix methods for topic modeling. This dissertation will investigate three popular techniques used in the data sciences: (i) structured matrix completion, (ii) supervision-aware dimensionality reduction, and (iii) dynamic topic modeling.

In Chapter 2, we describe our work [88] on iterative methods for *structured matrix completion.* This problem arises in many situations where the incomplete matrices admit additional structure in the missing entries, besides the low-rank structure of the whole matrix. One example of such structure in the missing entries is when the probability that an entry is observed or not depends mainly on the value of the entry. The question is whether we are able to better recover these matrices knowing that they admit this additional structure. In recent work [118], a modification to the standard nuclear norm minimization for matrix completion has been made to take into account *structural differences* between observed and unobserved entries. In our work [88], we propose adjusting an Iteratively Reweighted Least Squares (IRLS) algorithm for low-rank matrix completion to take into account *sparsity-based* structure in the missing entries. This structure is motivated by many applications in which the missing values tend to be near a certain value in the ℓ_0 or ℓ_1 norm sense. We also propose an iterative gradient-projection-based implementation of the algorithm, and present numerical experiments showcasing the performance of the algorithm compared to the standard IRLS algorithm in structured settings.

In Chapter 3, we describe our work [69] on semi-supervised nonnegative matrix factorization (SSNMF) for learning tasks which use utilize information divergence as an error function. We provide motivation for these models as maximum likelihood estimators. Further, we show, as in [104], that our proposed models generalize nonnegative matrix factorization (NMF) to supervised learning tasks and provide a topic model which simultaneously provides a lower dimensional representation of the data and a predictive model for targets. The Poisson distribution is particularly well suited for integer-valued datasets, such as documents represented by a vector of term frequencies [135], which leads to the information divergence in the MLE model [37, 74, 121]. We derive training methods using multiplicative updates for each new model, and demonstrate the application of these models to document classification, particularly on the 20 Newsgroups dataset.

In Chapter 4, we describe our work [89] on dynamic topic modeling for temporal text datasets using nonnegative tensor decomposition. Temporal data, such as a news articles or Twitter feeds, often consists of a mixture of long-lasting trends and popular but short-lasting topics of interest. A truly successful topic modeling strategy should be able to detect both types of topics and clearly locate them in time. We show that nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD) successfully detects such short-lasting topics that other popular methods such as latent Dirichlet allocation (LDA) and nonnegative matrix factorization (NMF) fail to fully detect. We demonstrate the ability of NCPD to discover short and long-lasting temporal topics in semi-synthetic and real-world data including news headlines and COVID-19 related tweets.

Chapter 2

An Iterative Method for Structured Matrix Completion

2.1 Introduction

Matrix completion is the task of filling-in, or predicting, the missing entries of a partially observed matrix from a subset of known entries. In today's data-driven world, data completion is essential, whether it is the main goal as in recommender systems, or a pre-processing step for other tasks like regression or classification. One popular example of a data completion task is the *Netflix Problem* [10, 11, 96], which was an open competition for the best collaborative filtering algorithm to predict unseen user ratings for movies. Given a subset of user-movie ratings, the goal is to predict the remaining ratings, which can be used to decide whether a certain movie should be recommended to a user. The Netflix Problem can be viewed as a matrix completion problem where the rows represent users, the columns represent movies, and the entries of the matrix are the corresponding user-movie ratings, most of which are missing.

Matrix completion problems are generally ill-posed without some additional information, since the missing entries could be assigned arbitrary values. In many instances, the matrix we wish to recover is known to be low-dimensional in the sense that it is low-rank, or approximately low-rank. For instance, a data matrix of all user-ratings of movies may be approximately low-rank because it is commonly believed that only a few factors contribute to an individual's tastes or preferences [24]. Low-rank matrix completion is a special case of the *affine rank minimization problem*, which arises often in machine learning, and is known to be NP-hard [49, 139].

Standard matrix completion strategies typically assume that there are no *structural differences* between observed and missing entries, which is an unrealistic assumption in many settings. Recent works [35, 118, 142, 147, 154] address various notions of the problem of structured matrix completion. General notions of structural difference include any setting in which whether an entry is observed or unobserved does not occur uniformly at random. For example, the probability that an entry is observed could depend not only on the value of that entry, but also on its location in the matrix. For instance, certain rows (or columns) may have substantially more entries than a typical row (or column); this happens in the Netflix Problem for very popular movies or so-called “super-users”.

In [118], Molitor and Needell propose a modification to the *standard nuclear norm minimization* for matrix completion to take into account structure when the submatrix of unobserved entries is sparse, or when the unobserved entries have lower magnitudes than the observed entries [118]. In our work, we focus on this notion of structure, in which the probability that an entry is observed or not depends mainly on the value of the entry. In particular, we are interested in sparsity-based structure in the missing entries, whereby the submatrix of missing values is close to 0 in the L_1 or L_0 norm sense. This is motivated by many situations in which the missing values tend to be near a certain value. For instance, missing data in chemical measurements might indicate that the measurement value is lower than the limit of detection of the device, and thus a typical missing measurement is smaller in value than a typical observed measurement. Similarly, in medical survey data, patients are more likely to respond to questions addressing noticeable symptoms, whereas a missing response may indicate a lack of symptoms [118]. In the Netflix problem, a missing rating of a movie might indicate the user’s lack of interest in that movie, thus suggesting a lower rating than otherwise expected. More generally, in survey data, incomplete data may be irrelevant or unimportant to the individual, therefore suggesting structure in the missing observations [118]. For an example in the setting of sensor networks, suppose we are given partial information about the signal strength between sensors, where the signal strength reading is inversely proportional to the distance between two sensors [24], and we would like to impute the missing signal strength readings. Signals may be missing because of low signal strength, indicating that perhaps these sensors are far from each other (or there are geographic obstacles between them). Thus, we obtain a partially observed matrix with structured observations—missing entries tend to have lower signal strength. Sensor networks give a low-rank matrix completion problem, more specifically of rank equal to two if the sensors are located in a plane, or three if they are located in three-dimensional space [108, 153]. Therefore, in these settings, we expect that the missing entries admit a sparsity-based structure in the L_1 norm sense.

2.1.1 Background and Related Work

The *Affine Rank Minimization Problem* (ARMP), or the problem of finding the minimum rank matrix in an affine set, is expressed as

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & \operatorname{rank}(\mathbf{X}) \\ \text{subject to} \quad & \mathscr{A}(\mathbf{X}) = b, \end{aligned} \tag{2.1}$$

where matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the optimization variable, $\mathscr{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^q$ is a linear map, and $b \in \mathbb{R}^q$ denotes the measurements. The affine rank minimization problem arises frequently in applications like system identification and control [110], collaborative filtering, low-dimensional Euclidean embeddings [50], sensor networks [15, 146, 152], quantum state tomography [64, 65], signal processing, and image processing.

Many algorithms have been proposed for ARMP, e.g. reweighted nuclear norm minimization [116], Singular Value Thresholding (SVT) [21], Fixed Point Continuation Algorithm (FPCA) [112], Iterative Hard Thresholding (IHT) [59], Optspace [90], Singular Value Projection (SVP) [82], Atomic Decomposition for Minimum Rank Approximation (AdMiRA) [105], Alternating Minimization approach [83], and the accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems (NNLS) [157], etc.

The low-rank matrix completion problem can be formulated as follows [26, 138]. Suppose we are given some matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ with a set Ω of partially observed entries, of size $|\Omega| \ll mn$. The goal is to recover the missing elements in $\mathbf{X}$. The low-rank matrix completion problem is a special case of the affine rank minimization problem where the set of affine constraints restrict certain entries of the matrix $\mathbf{X}$ to equal observed values. In this case, the linear operator $\mathscr{A}$ is a sampling operator, and the problem can be written as

$$\begin{aligned} &\underset{\mathbf{X}}{\text{minimize}} \quad \operatorname{rank}(\mathbf{X}) \\ &\text{subject to} \quad \mathbf{X}_{ij} = \mathbf{M}_{ij}, \quad (i,j) \in \Omega, \end{aligned}$$

where $\mathbf{M}$ is the matrix we would like to recover, and where Ω denotes the set of entries which are revealed. We define the sampling operator $\mathscr{P}_\Omega(\mathbf{X}) : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ via

$$(\mathscr{P}_\Omega(\mathbf{X}))_{ij} = \begin{cases} \mathbf{X}_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega, \end{cases}$$

as in [26]. Further, Ω^c denotes the complement of Ω, i.e., all index pairs (i,j) that are not in Ω. Thus, Ω^c corresponds to the collection of missing entries. The *degrees of freedom ratio* of a partially observed $m \times n$ matrix of rank r is given by $FR = r(m+n-r)/|\Omega|$. Thus, the larger the degrees of freedom ratio is, the harder it becomes to recover the matrix M.

The rank minimization problem (2.1) is NP-hard in general, and therefore we consider its convex relaxation [23, 24, 26, 34, 49, 139],

$$\begin{aligned} &\underset{X}{\text{minimize}} \quad \|\mathbf{X}\|_* \\ &\text{subject to} \quad \mathscr{A}(\mathbf{X}) = b, \end{aligned} \tag{2.2}$$

where $\|\cdot\|_*$ denotes the nuclear norm, given by the sum of singular values.

Inspired by the iteratively reweighted least squares (IRLS) algorithm for sparse vector recovery analyzed in [43], iteratively reweighted least squares algorithms [54, 100, 117] have been proposed as a computationally efficient method for low-rank matrix recovery (see Section 2.2.3). Instead of minimizing the nuclear norm, the algorithms essentially minimize the Frobenius norm of the matrix, subject to affine constraints. Properly reweighting this norm produces low-rank solutions under suitable assumptions. In [117], Mohan and Fazel propose a family of Iterative Reweighted Least Squares algorithms for matrix rank minimization, called IRLS-p (for $0 \leq p \leq 1$), as a computationally efficient way to improve over the performance of nuclear norm minimization. In addition, a gradient projection algorithm is presented as an efficient implementation for the algorithm, which exhibits improved recovery when compared to existing algorithms.

Generally, standard matrix completion strategies assume that there are no structural differences between observed and unobserved entries. However, recent works [35, 36, 118, 137, 142, 147, 154] also address various notions of the problem of structured matrix completion in mathematical, statistical, and machine learning frameworks. In our work, we are interested in sparsity-based structure. This notion of structure was proposed in [118], where the standard nuclear norm minimization problem for low-rank matrix completion is modified to take into account sparsity-based structure by regularizing the values of the unobserved entries. We refer to this algorithm as Structured NNM (see Section 2.3.1).

2.1.2 Contribution

We adapt an iterative algorithm for low-rank matrix completion to take into account sparsity-based structure in unobserved entries by adjusting the IRLS-p algorithm studied in [117]. We refer to our algorithm as *Structured IRLS*. We also present a gradient-projection-based implementation, called *Structured sIRLS* (motivated by sIRLS in [117]). The main motivations for our approach, along with its advantages, are as follows:

- **Iterative algorithm for structured matrix completion.** Much work has been put into developing iterative algorithms (SVT [21], FPCA [112], IHT [59], IRLS [54, 100, 117], etc.) for ARMP, rather than solving the nuclear norm convex heuristic (NNM). We develop the first (to our knowledge) iterative algorithm that addresses the structured low-rank matrix completion problem, for which Structured NNM has been proposed. Indeed, iterative methods are well-known to offer ease of implementation and reduced computational resources, making our approach attractive in the large-scale settings.

- **Comparable performance with Structured NNM.** Structured NNM adapts nuclear norm minimization and ℓ_1 norm minimization, which are common heuristics for minimizing rank and inducing sparsity, respectively. For various structured regimes, we consider small-sized matrices and show that our proposed iterative method is comparable to Structured NNM on "hard" matrix completion problems and with "optimal" parameter choices for Structured NNM.

- **Improved IRLS recovery for structured matrices.** We show that in structured settings, Structured sIRLS often performs better than the sIRLS algorithm, as follows. We perform numerical experiments that consider $19^2 = 361$ combinations of different sampling rates of the zero and nonzero entries, in order to demonstrate various levels of sparsity in the missing entries. Consider for example Figure 2.1, on matrices of size 1000×1000 of rank 10, in which our proposed method outperforms standard sIRLS in over 90% of the structured experiments (and also in many of the unstructured experiments).

- **Handle hard problems.** We consider problems of varying degrees of freedom, and a priori rank knowledge. We show that Structured sIRLS often outperforms the sIRLS algorithm in structured settings for hard matrix completion problems, i.e. where the degrees of freedom ratio is greater than 0.4.

- **Handle noisy measurements.** We consider matrices with noisy measurements with two different levels of noise. We show that for small enough noise Structured sIRLS often performs better than sIRLS in structured settings. As the noise gets larger, both converge to the same performance.

Our implementations of Structured-sIRLS and code for reproducing experiments are publicly available[1].

[1] https://github.com/lara-kassab/structured-matrix-completion-IRLS

2.1.3 Organization

We review related iteratively reweighted least squares algorithms for recovering sparse vectors and low-rank matrices in Section 2.2. In Section 2.3, we describe the structured matrix completion problem, propose for this problem an iterative algorithm, Structured IRLS, and present preliminary analytic remarks. Furthermore, we present a computationally efficient implementation, Structured sIRLS. In Section 2.4, we run numerical experiments to showcase the performance of this method, and compare it to the performance of sIRLS and Structured NNM on various structured settings.

2.2 Iteratively Reweighted Least Squares Algorithms

In this section, we set notation for the rest of the chapter, and we review related algorithms for recovering sparse vectors and low-rank matrices.

2.2.1 Notation

We denote vectors with lowercase letters x and matrices with uppercase boldface letters $\mathbf{X}$. The entries of a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ are denoted by x_{ij} or $\mathbf{X}_{ij}$ (the entry in row i and column j of $\mathbf{X}$). Let $\mathbf{I}$ denote the identity matrix and $\mathbf{1}$ the vector of all ones. The *trace* of a square matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$ is the sum of its diagonal entries, and is denoted by $\mathrm{Tr}(\mathbf{X}) = \sum_{i=1}^{m} x_{ii}$. We denote the adjoint matrix of $\mathbf{X}$ by $\mathbf{X}^* \in \mathbb{R}^{n \times m}$. Without loss of generality, we assume $m \leq n$ and we write the singular value decomposition of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*.$$

Here $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are unitary matrices, and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_m) \in \mathbb{R}^{m \times n}$ is a diagonal matrix, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ are the *singular values*. The *rank* of $\mathbf{X} \in \mathbb{R}^{m \times n}$, denoted by $\mathrm{rank}(\mathbf{X})$, equals the number of nonzero singular values of $\mathbf{X}$. Further, the *Frobenius norm* of the matrix $\mathbf{X}$ is defined by

$$\|\mathbf{X}\|_F = \sqrt{\mathrm{Tr}(\mathbf{X}\mathbf{X}^*)} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2\right)^{1/2} = \left(\sum_{i=1}^{m}\sigma_i^2\right)^{1/2}.$$

The *nuclear norm* of a matrix $\mathbf{X}$ is defined by $\|\mathbf{X}\|_* = \sum_{i=1}^{m}\sigma_i$. Given a vector $w \in \mathbb{R}^n$ of positive weights, we define the *weighted ℓ_2 norm* of a vector $z \in \mathbb{R}^n$ as

$$\|z\|_{\ell_2(w)} = \left(\sum_{i=1}^{n} w_i z_i^2\right)^{1/2}.$$

Let $z(\mathbf{X})$ denote the vector of missing entries of a matrix $\mathbf{X}$, and let $z^2(\mathbf{X})$ denote the corresponding vector with entries squared, i.e. $z^2(\mathbf{X}) = z(\mathbf{X}) \odot z(\mathbf{X})$ where $\odot$ denotes elementwise multiplication.

2.2.2 Sparse Vector Recovery

Given a vector x, the value $\|x\|_0$ denotes the number of nonzero entries of x, and is known as the *ℓ_0 norm* of x. The sparse vector recovery problem is described as

$$\begin{aligned} \text{minimize} \quad & \|x\|_0, \\ \text{subject to} \quad & \mathbf{A}x = b, \end{aligned} \tag{2.3}$$

where $x \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{m\times n}$. This problem is known to be NP-hard. A commonly used convex heuristic for this problem is ℓ_1 minimization [25, 46],

$$\begin{aligned} \text{minimize} \quad & \|x\|_1, \\ \text{subject to} \quad & \mathbf{A}x = b. \end{aligned} \tag{2.4}$$

Indeed, many algorithms for solving (2.3) and (2.4) have been proposed. In [43], Daubechies et al. propose and analyze a simple and computationally efficient reweighted algorithm for sparse vector recovery, called the Iterative Reweighted Least Squares algorithm, IRLS-p, for any $0 < p \leq 1$. Its k-th iteration is given by

$$x^{k+1} = \operatorname*{argmin}_{x} \left\{ \sum_{i=1}^{n} w_i^k x_i^2 \ : \ \mathbf{A}x = b \right\},$$

where $w^k \in \mathbb{R}^n$ is a weight vector with $w_i^k = \left(|x_i^k|^2 + \epsilon_k^2\right)^{p/2-1}$, and where $\epsilon_k > 0$ is a regularization parameter added to ensure that w^k is well-defined. For $p = 1$, [43] gives a theoretical guarantee for sparse recovery similar to ℓ_1 norm minimization.

2.2.3 Low-rank Matrix Recovery

We review two related algorithms [54, 117] for low-rank matrix recovery that generalize the iteratively reweighted least squares algorithm analyzed in [43] for sparse vector recovery. In general, minimizing the Frobenius norm subject to affine constraints does not lead to low-rank solutions; however, properly reweighting this norm produces low-rank solutions under suitable assumptions [54, 117].

In [54], Fornasier et al. propose a variant of the reweighted least squares algorithm for sparse recovery for nuclear norm minimization (or low-rank matrix recovery), called IRLS-M. The k-th iteration of IRLS-M is given by

$$\mathbf{X}^{k+1} = \operatorname*{argmin}_{\mathbf{X}} \left\{ \|(\mathbf{W}^k)^{1/2}\mathbf{X}\|_F^2 : \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}) \right\}. \tag{2.5}$$

Here $\mathbf{W}^k \in \mathbb{R}^{m\times m}$ is a weight matrix defined as $\mathbf{W}^0 = \mathbf{I}$, and for $k > 0$, $\mathbf{W}^k = \mathbf{U}^k(\mathbf{\Sigma}_{\epsilon_k}^k)^{-1}(\mathbf{U}^k)^*$, where $\mathbf{X}^k(\mathbf{X}^k)^* = \mathbf{U}^k(\mathbf{\Sigma}^k)^2(\mathbf{U}^k)^*$ and $\mathbf{\Sigma}_{\epsilon_k} = \text{diag}(\max\{\sigma_j, \epsilon_k\})$. Indeed, each iteration of (2.5) minimizes a weighted Frobenius norm of the matrix X. Under the assumption that the linear measurements fulfill a suitable generalization of the *Null Space Property* (NSP), the algorithm is guaranteed to iteratively recover any matrix with an error on the order of the best rank k approximation [54]. The algorithm essentially has the same recovery guarantees as nuclear norm minimization. Though the Null Space Property fails in the matrix completion setup, the authors illustrate numerical experiments which show that the IRLS-M algorithm still works very well in this setting for recovering low-rank matrices. Further, for the matrix completion problem, the algorithm takes advantage of the Woodbury matrix identity, allowing an expedited solution to the least squares problem required at each iteration [54].

In [117], Mohan and Fazel propose a related family of Iterative Reweighted Least Squares algorithms for matrix rank minimization, called IRLS-p (for $0 \leq p \leq 1$), as a computationally efficient way to improve over the performance of nuclear norm minimization. The k-th iteration of IRLS-p is given by

$$\mathbf{X}^{k+1} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \operatorname{Tr}(\mathbf{W}_p^k \mathbf{X}^\top \mathbf{X}) : \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}) \right\}, \tag{2.6}$$

where $\mathbf{W}_p^k \in \mathbb{R}^{m \times m}$ is a weight matrix defined as $\mathbf{W}_p^0 = \mathbf{I}$, and for $k > 0$, $\mathbf{W}_p^k = ((\mathbf{X}^k)^\top \mathbf{X}^k + \gamma^k \mathbf{I})^{\frac{p}{2}-1}$. Here $\gamma^k > 0$ is a regularization parameter added to ensure that $\mathbf{W}_p^k$ is well-defined. Each iteration of (2.6) minimizes a weighted Frobenius norm of the matrix X, since

$$\operatorname{Tr}(\mathbf{W}_p^{k-1} \mathbf{X}^\top \mathbf{X}) = \|(\mathbf{W}_p^{k-1})^{1/2} \mathbf{X}\|_F^2.$$

The algorithms can be viewed as (locally) minimizing certain smooth approximations to the rank function. When $p = 1$, theoretical guarantees are given similar to those for nuclear norm minimization, i.e., recovery of low-rank matrices under the assumptions that the operator defining the constraints satisfies a specific *Null Space Property*. Further, for $p < 1$, IRLS-p shows better empirical performance in terms of recovering low-rank matrices than nuclear norm minimization. In addition, a gradient projection algorithm, IRLS-GP, is presented as an efficient implementation for IRLS-p. Further, this same paper presents a related family of algorithms sIRLS-p (or short IRLS), which can be seen as a first-order method for locally minimizing a smooth approximation to the rank function. The results exploit the fact that these algorithms can be derived from the KKT conditions for minimization problems whose objectives are suitable smooth approximations to the rank function [117]. We will sometimes refer to IRLS-p (resp. sIRLS-p) studied in [117] as IRLS (resp. sIRLS).

The algorithms proposed in [54, 117] differ mainly in their implementations, and in the update rules of the weights and their corresponding regularizers. In IRLS-M [54], the weights are updated as $\mathbf{W}^k = \mathbf{U}^k \operatorname{diag}(\max(\sigma_j^k, \epsilon_k)^{-1})(\mathbf{U}^k)^*$, and in IRLS-$p$ [117] they are updated as $\mathbf{W}^k = \mathbf{U}^k \operatorname{diag}(((\sigma_j^k)^2 + \gamma_k^2)^{-1/2})(\mathbf{U}^k)^*$. Further, each of the regularization parameters ϵ_k and γ_k are updated differently. The IRLS-M algorithm makes use of the rank of the matrix (either given or estimated), and thus the choice of parameter ϵ_k depends on this given or estimated rank. On the other hand, the IRLS-p algorithm chooses and updates its regularizer γ_k based on prior sensitivity experiments.

Terminology	
NNM	Nuclear Norm Minimization
Structured NNM	Adjusted NNM for sparsity-based structure in the missing entries, proposed in [118]
IRLS-p	Iterative Reweighted Least Squares algorithms for matrix rank minimization, proposed in [117]
sIRLS	short IRLS-p, proposed in [117]
Structured IRLS	Our proposed algorithm: adjusted IRLS for sparsity-based structure in the missing entries
Structured sIRLS	Our proposed implementation: adjusted sIRLS for sparsity-based structure in the missing entries

To re-iterate, we differ from IRLS-p in [117] by considering the (sparsity-based) structured matrix completion problem, and we differ from Structured NNM in [118] by considering an iterative approach to the problem.

2.3 Structured Iteratively Reweighted Least Squares Algorithms

In this section, we first introduce the structured matrix completion problem. Second, we introduce and analyze our proposed algorithm and implementation.

2.3.1 Problem Statement

In [118], the authors propose adjusting the standard nuclear norm minimization (NNM) strategy for matrix completion to account for structural differences between observed and unobserved entries. This could be achieved by adding to problem (2.2) a regularization term on the unobserved entries, which results in a semidefinite program:

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & \|\mathbf{X}\|_* + \alpha \|\mathscr{P}_{\Omega^c}(\mathbf{X})\| \\ \text{subject to} \quad & \mathscr{P}_{\Omega}(\mathbf{X}) = \mathscr{P}_{\Omega}(\mathbf{M}), \end{aligned} \tag{2.7}$$

where $\alpha > 0$, and where $\|\cdot\|$ is an appropriate matrix norm. If most of the missing entries are zero except for a few, then the ℓ_1 norm is a natural choice[2]. If the missing entries are mostly close to zero, then the ℓ_2 norm is a natural choice. The authors show that the proposed method outperforms nuclear norm minimization in certain structured settings. We refer to this method as Structured Nuclear Norm Minimization (Structured NNM).

Equation (2.7) very closely resembles the problem of decomposing a matrix into a low-rank component and a sparse component (see e.g. [29]). A popular method is Robust Principal Component Analysis (RPCA) [22], where one assumes that a low-rank matrix has some set of its entries corrupted. In a more recent paper [144], reweighted least squares optimization is applied to develop a novel online Robust PCA algorithm for sequential data processing. It would be interesting to also consider generalizations of Equation (2.7) in which the projection $\mathscr{P}_{\Omega}$ is allowed to be a more general affine transformation $\mathscr{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, and $\mathscr{P}_{\Omega^c}$ could be an affine transformation mapping into the nullspace of $\mathscr{A}$.

[2] The method can be rescaled if there instead is a preference for the missing entries to be near a nonzero constant.

2.3.2 Proposed Algorithm: Structured IRLS

We propose an iterative reweighted least squares algorithm related to [54, 117] for matrix completion with structured observations. In particular, we adjust the IRLS-p algorithm proposed in [117] to take into account the sparsity-based structure in the missing entries.

The k-th iteration of IRLS-p is given by

$$\mathbf{X}^k = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \|(\mathbf{W}_p^{k-1})^{1/2}\mathbf{X}\|_F^2 : \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}) \right\},$$

where $\mathbf{X}^k \in \mathbb{R}^{m\times n}$ denotes the k-th approximation of the true matrix $\mathbf{M}$, $\mathbf{W}_p^k \in \mathbb{R}^{m\times m}$ is a weight matrix defined as $\mathbf{W}_p^0 = \mathbf{I}$, and for $k > 0$, $\mathbf{W}_p^k = ((\mathbf{X}^k)^\top \mathbf{X}^k + \gamma^k \mathbf{I})^{\frac{p}{2}-1}$. Here $\gamma^k > 0$ is a regularization parameter added to ensure that $\mathbf{W}_p^k$ is well-defined.

We adjust IRLS-p by adding a regularization term on the unobserved entries in each iteration, namely a weighted ℓ_2 norm as proposed in [43] to induce sparsity. We define the corresponding weights at the k-th iteration as $w_q^k = (z^2(\mathbf{X}^k) + \epsilon^k \mathbf{1})^{\frac{q}{2}-1}$, where $0 < \epsilon^k \leq \epsilon^{k-1}$, and $0 \leq q \leq 1$. Here $z(\mathbf{X}^k)$ denotes the vector of missing entries of the the k-th approximation $\mathbf{X}^k$, and recall that $z^2(\mathbf{X}^k)$ denotes the vector with entries squared. The algorithm is then designed to promote low-rank structure in the recovered matrix with sparsity in the missing entries at each iteration. We give a description of the choice of parameters in Section 2.4.1. We refer to the algorithm as *Structured IRLS*; it is outlined in Algorithm 1. Note that each iteration of Structured IRLS solves a quadratic program, and for $\alpha = 0$, the algorithm reduces to IRLS-p studied in [117].

In many applications, missing values tend to be near a certain value, e.g. the maximum possible value in the range, or alternatively the lowest possible value (“1 star" in movie ratings). In cases where this value is nonzero, our objective function can be adjusted accordingly. For example, one can shift the given entries to be $a - \mathbf{M}_{ij}$ for all i, j, where $a \in \mathbb{R}$ is the constant or threshold of interest.

Algorithm 1: Structured IRLS for Matrix Completion

```
input     : $\mathscr{P}_\Omega$, $\mathbf{M}$
set       : $k = 1$, $\alpha > 0$, and $0 \leq p, q \leq 1$
initialize: $\mathbf{X}^0 = \mathscr{P}_\Omega(\mathbf{M})$, $\mathbf{W}_p^0 = \mathbf{I}$, $w_q^0 = \mathbf{1}$, $\gamma^1 > 0$, $\epsilon^1 > 0$
while not converged do
    $\mathbf{X}^k = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \|(\mathbf{W}_p^{k-1})^{1/2}\mathbf{X}\|_F^2 + \alpha \|z(\mathbf{X})\|^2_{\ell_2(w_q^{k-1})} \ : \ \mathscr{P}_\Omega(\mathbf{X}) \ = \ \mathscr{P}_\Omega(\mathbf{M}) \right\}$
    compute: $\mathbf{W}_p^k = ((\mathbf{X}^k)^\top \mathbf{X}^k + \gamma^k \mathbf{I})^{\frac{p}{2}-1}$ and $w_q^k = (z^2(\mathbf{X}^k) + \epsilon^k \mathbf{1})^{\frac{q}{2}-1}$
    update : $0 < \gamma^{k+1} \leq \gamma^k$, $0 < \epsilon^{k+1} \leq \epsilon^k$
    set    : $k = k + 1$
end
```

Each iteration of Structured IRLS solves a quadratic program. The algorithm can be adjusted to have the ℓ_2 norm (instead of the weighted ℓ_2 norm) for the regularization term on the unobserved entries by fixing the weights $w_q^k = \mathbf{1}$. Further, we can impose nonnegativity constraints on the missing entries by thresholding all missing entries to be nonnegative.

We now provide an analytic remark, similar to [118, Proposition 1], applied to the objective functions for each iteration of IRLS [117] and Structured IRLS. We consider the simplified setting in which all of the unobserved entries are exactly zero. We show that the approximation given by an iteration of Structured IRLS will always perform at least as well as that of IRLS with the same weights assigned. This remark is weaker than [118, Proposition 1] as it does not apply to the entire algorithm; instead it only bounds the performance of a single iterative step.

Remark 2.3.1. Let

$$\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \|\mathbf{W}^{1/2}\mathbf{X}\|_F^2 \ : \ \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}) \right\}$$

be the minimizer of the objective function of each iterate in IRLS [117]. Let

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \|\mathbf{W}^{1/2}\mathbf{X}\|_F^2 + \alpha \|\mathscr{P}_{\Omega^c}(\mathbf{X})\|^2 \ : \ \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}) \right\}$$

be the minimizer of the objective function generalizing[3] each iterate in Structured IRLS (with $\alpha > 0$). If $\mathscr{P}_{\Omega^c}(\mathbf{M})$ is the zero matrix and the same weights $\mathbf{W}$ are assigned, then $\|\mathbf{M}-\hat{\mathbf{X}}\| \leq \|\mathbf{M}-\tilde{\mathbf{X}}\|$ for any matrix norm $\|\cdot\|$.

Proof. By definition of $\hat{\mathbf{X}}$, we have $\|\mathbf{W}^{1/2}\hat{\mathbf{X}}\|_F^2 + \alpha\|\mathscr{P}_{\Omega^c}(\hat{\mathbf{X}})\|^2 \leq \|\mathbf{W}^{1/2}\tilde{\mathbf{X}}\|_F^2 + \alpha\|\mathscr{P}_{\Omega^c}(\tilde{\mathbf{X}})\|^2$. Similarly, by definition of $\tilde{\mathbf{X}}$, we have $\|\mathbf{W}^{1/2}\tilde{\mathbf{X}}\|_F^2 \leq \|\mathbf{W}^{1/2}\hat{\mathbf{X}}\|_F^2$. Therefore,

$$\begin{aligned}\|\mathbf{W}^{1/2}\hat{\mathbf{X}}\|_F^2 + \alpha\|\mathscr{P}_{\Omega^c}(\hat{\mathbf{X}})\|^2 &\leq \|\mathbf{W}^{1/2}\tilde{\mathbf{X}}\|_F^2 + \alpha\|\mathscr{P}_{\Omega^c}(\tilde{\mathbf{X}})\|^2 \\ &\leq \|\mathbf{W}^{1/2}\hat{\mathbf{X}}\|_F^2 + \alpha\|\mathscr{P}_{\Omega^c}(\tilde{\mathbf{X}})\|^2.\end{aligned}$$

Since $\alpha > 0$, this implies $\|\mathscr{P}_{\Omega^c}(\hat{\mathbf{X}})\|^2 \leq \|\mathscr{P}_{\Omega^c}(\tilde{\mathbf{X}})\|^2$. We have

$$\begin{aligned}\|\mathbf{M}-\hat{\mathbf{X}}\| &= \|\mathscr{P}_{\Omega^c}(\hat{\mathbf{X}})\| && \text{since } \mathscr{P}_{\Omega}(\mathbf{M}) = \mathscr{P}_{\Omega}(\hat{\mathbf{X}}) \text{ and } \mathscr{P}_{\Omega^c}(\mathbf{M}) = 0 \\ &\leq \|\mathscr{P}_{\Omega^c}(\tilde{\mathbf{X}})\| \\ &= \|\mathbf{M}-\tilde{\mathbf{X}}\| && \text{since } \mathscr{P}_{\Omega}(\mathbf{M}) = \mathscr{P}_{\Omega}(\tilde{\mathbf{X}}) \text{ and } \mathscr{P}_{\Omega^c}(\mathbf{M}) = 0.\end{aligned}$$

□

2.3.3 Proposed Implementation: Structured sIRLS

In this section, we propose a gradient-projection-based implementation of Structured IRLS, that we will refer to as *Structured sIRLS*. Indeed, sIRLS stands for short IRLS (in analogy to [117]), the reason being we do not perform gradient descent until convergence; instead we take however many steps desired. Further, calculating $\mathscr{P}_{\Omega}(\mathbf{X})$ is computationally cheap, so the gradient projection algorithm can be used to efficiently solve the quadratic program in each iteration of Structured IRLS.

In this implementation, we do not perform projected gradient descent on

[3] Here $\|\cdot\|$ is an arbitrary matrix norm; one recovers Structured IRLS by choosing the norm $\ell_2(w)$.

$$\|(\mathbf{W}_p^{k-1})^{1/2}\mathbf{X}\|_F^2 + \alpha\|z(\mathbf{X})\|_{\ell_2(w_q^{k-1})}^2,$$

with $\mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M})$ for each iteration k. Instead, we perform projected gradient descent on $\|z(\mathbf{X})\|_{\ell_2(w_q^{k-1})}^2$ and $\|(\mathbf{W}_p^{k-1})^{1/2}\mathbf{X}\|_F^2$ consecutively. This allows us to update the weights before each alternating step, and to control how many gradient steps we would like to perform on each function.

We follow [117] for the derivation of the gradient step of $\|(\mathbf{W}_p^{k-1})^{1/2}\mathbf{X}\|_F^2$ at the k-th iteration. Indeed, we consider the smooth Schatten-p function, for $p > 0$:

$$f_p(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I})^{\frac{p}{2}} = \sum_{i=1}^{n}\left(\sigma_i^2(\mathbf{X}) + \gamma\right)^{\frac{p}{2}}.$$

Note that $f_p(\mathbf{X})$ is differentiable for $p > 0$, and convex for $p \geq 1$ [117]. For $\gamma = 0$ we have $f_1(\mathbf{X}) = \|\mathbf{X}\|_*$, which is also known as the Schatten-1 norm. Again for $\gamma = 0$, we have $f_p(\mathbf{X}) \to \mathrm{rank}(\mathbf{X})$ as $p \to 0$ [117]. Further, for $p = 0$, we define

$$f_0(x) = \log\det(\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I}),$$

a smooth surrogate for $\mathrm{rank}(\mathbf{X}^\top\mathbf{X})$ (see e.g. [49, 50, 117, 139]). Thus, it is of interest to minimize $f_p(\mathbf{X})$ subject to the set of constraints $\mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M})$ on the observed entries.

The gradient projection iterates of Structured sIRLS are given by

$$\mathbf{X}^{k+1} = \mathscr{P}_{\Omega^c}(\mathbf{X}^k - s^k\nabla f_p(\mathbf{X}^k)) + \mathscr{P}_\Omega(\mathbf{M}),$$

where s^k denotes the gradient step size at the k-th iteration and $\nabla f_p(\mathbf{X}^k) = \mathbf{X}^k\mathbf{W}_p^k$, where we iteratively define $\mathbf{W}_p^k$ as

$$\mathbf{W}_p^k = (\mathbf{X}^{k^\top}\mathbf{X}^k + \gamma^k\mathbf{I})^{\frac{p}{2}-1},$$

with $0 < \gamma^k \leq \gamma^{k-1}$. This iterate describes our gradient step promoting low-rankness, where we preserve the observed entries and update only the missing entries.

Further, we promote sparsity in the missing entries as follows. Instead of minimizing the ℓ_1 norm of the vector of missing entries, we iteratively minimize a re-weighted ℓ_2 norm of missing entries as described in [43]. Let $z(\mathbf{X}^k)$ denote the vector of missing entries of the the k-th approximation $\mathbf{X}^k$. Define the weighted ℓ_2 norm of $z(\mathbf{X})$ as

$$g_q(\mathbf{X}) = \|z(\mathbf{X})\|^2_{\ell_2(w_q)} = \sum_{i=1}^{mn-|\Omega|} (w_q)_i z_i^2(\mathbf{X}),$$

where $(w_q)_i = (z_i^2(\mathbf{X}) + \epsilon)^{q/2-1}$ (as done in [43]). The i-th entry of the gradient of $g_q(\mathbf{X})$ is given by $(\nabla g_q(\mathbf{X}))_i = 2(w_q)_i z_i$. Therefore, the gradient projection iterates are given by

$$z(\mathbf{X}^{k+1}) = z(\mathbf{X}^k) - c^k \nabla g_q(\mathbf{X}^k),$$

where c^k denotes the gradient step size at the k-th iteration. We iteratively define the weights w_q^k as

$$w_q^k = (z^2(\mathbf{X}^k) + \epsilon^k \mathbf{1})^{\frac{q}{2}-1},$$

where $0 < \epsilon^k \le \epsilon^{k-1}$. We outline in Algorithm 2 Structured sIRLS, a gradient-projection-based implementation of Structured IRLS.

A rank estimate r of the matrix $\mathbf{M}$ is used as an input to truncate the singular value decomposition (SVD) when computing the weights $\mathbf{W}_p^k$. In our implementation, we use a randomized algorithm for SVD computation [70] to reduce the computational complexity. For example, consider finding the r dominant components of the singular value decomposition of an $m \times n$ matrix. For a dense input matrix, randomized algorithms require $O(mn\log(r))$ floating-point operations in contrast with $O(mnr)$ for classical algorithms [70]. When the rank of the matrix is not estimated or provided, we instead choose r to be $\min\{r_{max}, \hat{r}\}$ at each iteration, where $\hat{r}$ is the largest integer such that $\sigma_{\hat{r}}(\mathbf{X}^k) > 10^{-2} \cdot \sigma_1(\mathbf{X}^k)$, and where $r_{max} = \left\lceil n\left(1 - \sqrt{1 - \frac{|\Omega|}{mn}}\right)\right\rceil$ (as implemented in [117]).

Algorithm 2: Structured sIRLS for Matrix Completion

```
input: $\mathscr{P}_\Omega$, $\mathbf{M}$, $r$
set    : $k = 1$, $0 \le p, q \le 1$, $k_s > 0$, $k_l > 0$, $c^k > 0$, $s^k > 0$
initialize: $X^0 = \mathscr{P}_\Omega(\mathbf{M})$, $w_q^0 = \mathbf{1}$, $\gamma^1 > 0$, $\epsilon^1 > 0$
while not converged do
    perform : take $k_s$ steps promoting sparsity, $z(\mathbf{X}^k) = z(\mathbf{X}^{k-1}) - c^k({w_q}^{k-1} \odot z(\mathbf{X}^{k-1}))$
    update  : update the weights promoting low-rankness, $\mathbf{W}_p^k = (\mathbf{X}^{k\top}\mathbf{X}^k + \gamma^k \mathbf{I})^{\frac{p}{2}-1}$
    perform : take $k_l$ steps promoting low-rankness, $\mathbf{X}^{k+1} = P_{\Omega^c}(\mathbf{X}^k - s^k \mathbf{X}^k \mathbf{W}_p^k) + P_\Omega(\mathbf{M})$
    update  : update the weights promoting sparsity, $w_q^k = (z^2(\mathbf{X}^{k+1}) + \epsilon^k \mathbf{1})^{\frac{q}{2}-1}$
    update  : update the regularizers, $0 < \gamma^{k+1} \le \gamma^k$, $0 < \epsilon^{k+1} \le \epsilon^k$
    set     : set $k = k + 1$
end
```

2.4 Numerical Experiments

In this section, we run numerical experiments to evaluate the performance of Structured sIRLS. We compare Structured sIRLS to the performance of sIRLS (studied in [117]) and Structured NNM (studied in [118]) on structured settings. Our code for Structured sIRLS is available at [87]. Further, we use the publicly available code of sIRLS [115].

First, in Section 2.4.1, we explain the choice of parameters we use. We describe our experiments for exact matrix completion in Section 2.4.2. For problems of varying degrees of difficulty in terms of the sampling rate, degrees of freedom, and sparsity levels, we find that Structured sIRLS often outperforms sIRLS and Structured NNM in the structured setting. In Section 2.4.3 we consider matrix completion with noise, finding that Structured sIRLS improves upon sIRLS in the structured setting with low noise. As the noise level increases, the performance of Structured sIRLS remains controlled, approximately the same as the performance of sIRLS.

2.4.1 Choice of Parameters

In all the numerical experiments, we adopt the same parameters. However, one can use different choices for parameters, or optimize some of the parameters. We normalize the input data matrix M to have a spectral norm of 1 (as done in [117]).

We are particularly interested in the case $p = q = 1$. In our experiments, we set $p = q = 1$, but generally, these parameters can be varied over the range $0 \leq p, q \leq 1$. Each value of p and q define a different objective function (see Sections 2.2.2 and 2.2.3 where q is referred to as p).

For the implementation parameters, we set $k_s = 1$ and $k_l = 10$, which means that we take one gradient step to promote sparsity and ten gradient steps to promote low-rankness, respectively. These parameters can be varied based on the low-rankness of the matrix and on the expected sparsity of its missing entries: increasing k_s promotes sparsity and increasing k_l promotes low-rankness. Further, we set the regularizers $\gamma^k = (1/2)^k$ and $\epsilon^k = (9/10)^k$ at the k-th iteration. Exponentially decreasing geometric sequences converging to 0 can be chosen for convenience. However, there are other possible choices for these regularizers, for example ϵ^k could depend on the $(s+1)$-th largest value of $z(\mathbf{X}^k)$, where s is the sparsity of $z(\mathbf{X}^k)$ (as done in [43]). Similarly, γ^k could depend on the $(r+1)$-th singular value of $\mathbf{X}^k$, where r is the rank of $\mathbf{M}$ (as done in [54]).

Lastly, for all k we set the step size $s^k = (\gamma^k)^{1-\frac{p}{2}}$ to promote low-rankness and $c^k = 10^{-6}$ to promote sparsity; however, these parameters could be scaled or varied. We define the relative distance between two consecutive approximations as

$$d(\mathbf{X}^k, \mathbf{X}^{k-1}) = \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F / \|\mathbf{X}^k\|_F.$$

We say the algorithm converges if we obtain $d(\mathbf{X}^k, \mathbf{X}^{k-1}) < 10^{-5}$. We set the tolerance 10^{-5} for both sIRLS and Structured sIRLS in our comparison experiments,[4] and we set the maximum number of iterations for Structured sIRLS to be 1000 and for sIRLS to be 5000.

[4] In the original implementation of sIRLS provided by the authors [115, 117], the tolerance value is set to 10^{-3}. However, Structured sIRLS converges much faster per iteration, thus attaining the tolerance 10^{-3} with fewer iterations. To report fair comparisons between the algorithms that do not overly benefit Structured sIRLS, we set the tolerance to 10^{-5} in addition to increasing the maximum number of iterations for sIRLS.

2.4.2 Exact Matrix Completion

We first investigate the performance of the Structured sIRLS algorithm when the observed entries are exact, i.e. there is no noise in the observed values. We construct $m \times n$ matrices of rank r as done in [118]. We consider $\mathbf{M} = \mathbf{M}_L\mathbf{M}_R$, where $\mathbf{M}_L \in \mathbb{R}^{m\times r}$ and $\mathbf{M}_R \in \mathbb{R}^{r\times n}$ are sparse matrices. Indeed, the entries of $\mathbf{M}_L$ (resp. $\mathbf{M}_R$) are chosen to be zero uniformly at random so that on average 70% (resp. 50%) of its entries are zero. The remaining nonzero entries are uniformly distributed at random between zero and one. The sparsity level of the resulting matrix $\mathbf{M}$ cannot be calculated exactly from the given sparsity levels of $\mathbf{M}_L$ and $\mathbf{M}_R$. Thus for each of the following numerical simulations, we indicate the average sparsity level of $\mathbf{M}$ (we refer to the density of $\mathbf{M}$ as the fraction of nonzero entries).

For each experiment with m, n, and r fixed, we choose twenty random matrices of the form $\mathbf{M} = \mathbf{M}_L\mathbf{M}_R$. We subsample from the zero and nonzero entries of the data matrix at various rates to generate a matrix with missing entries. We define the *relative error of Structured sIRLS* as

$$\|\mathbf{M} - \hat{\mathbf{X}}\|_F / \|\mathbf{M}\|_F,$$

where $\hat{\mathbf{X}}$ is the output of the Structured sIRLS algorithm. Similarly, we define the *relative error of sIRLS* as

$$\|\mathbf{M} - \tilde{\mathbf{X}}\|_F / \|\mathbf{M}\|_F,$$

where $\tilde{X}$ is the output of the sIRLS algorithm. The *average ratio* is then defined as

$$\|\mathbf{M} - \hat{\mathbf{X}}\|_F / \|\mathbf{M} - \tilde{\mathbf{X}}\|_F.$$

We say Structured sIRLS outperforms sIRLS when the average ratio is less than one, and vice versa when the average ratio is greater than or equal to one. These two cases, when the average ratio is strictly less than or greater than or equal to one, are visually represented by the white and black squares, respectively, in the bottom right plots of Figures 2.1–2.3 and 2.5. We refer to this binary notion of average ratio as *binned average ratio*. We report each of these error values in our numerical experiments.

It is important to note that the setting we are interested in is the structured setting where the submatrix of missing values is close to 0 in the L_1 or L_0 norm sense. This setting can be observed in the upper left triangle of the images in Figures 2.1–2.7 (in particular, this is the region above the diagonal gray line in the bottom rows of Figures 2.1–2.7). In this upper-left triangular region, the percentage of nonzero entries that are sampled is greater than the percentage of zero entries that are sampled. Hence the region above the diagonal gray lines is the structured setting that Structured sIRLS is designed for.

In general, algorithms obtain better accuracy as we move right along a row or up along a column in Figures 2.1–2.7, since we are sampling more and more entries. In addition, it is important to note that in all experiments we are using the same algorithm (with fixed parameters) for all the cases considered in our computations, without any parameter optimization. The Structured sIRLS algorithm promotes sparsity in all the cases, even in the unstructured settings. Omitting the sparsity promoting step would result in an algorithm promoting only low-rankness.

1000×1000 **rank 10 matrices**

In Figure 2.1, we construct twenty random matrices of size 1000×1000 and of rank 10, as described in Section 2.4.2. Error ratios below one, in the bottom left plot of Figure 2.1, indicate that Structured sIRLS outperforms sIRLS. In this particular experiment, we observe that Structured sIRLS outperforms sIRLS for most of the structured cases (the upper left triangle above the gray line), and more. For this particular experiment, it turns out that this happens roughly when the decimal percentage of sampled nonzero entries is greater than 0.2.

Note that in the case where all entries are observed (no longer a matrix completion problem), both relative errors are 0 and thus the average ratio is 1. We only say that Structured sIRLS outperforms sIRLS when the average ratio is strictly less than 1, and this is why the upper right pixel in the bottom right plot of Figure 2.1 is black. The same is true in later figures.

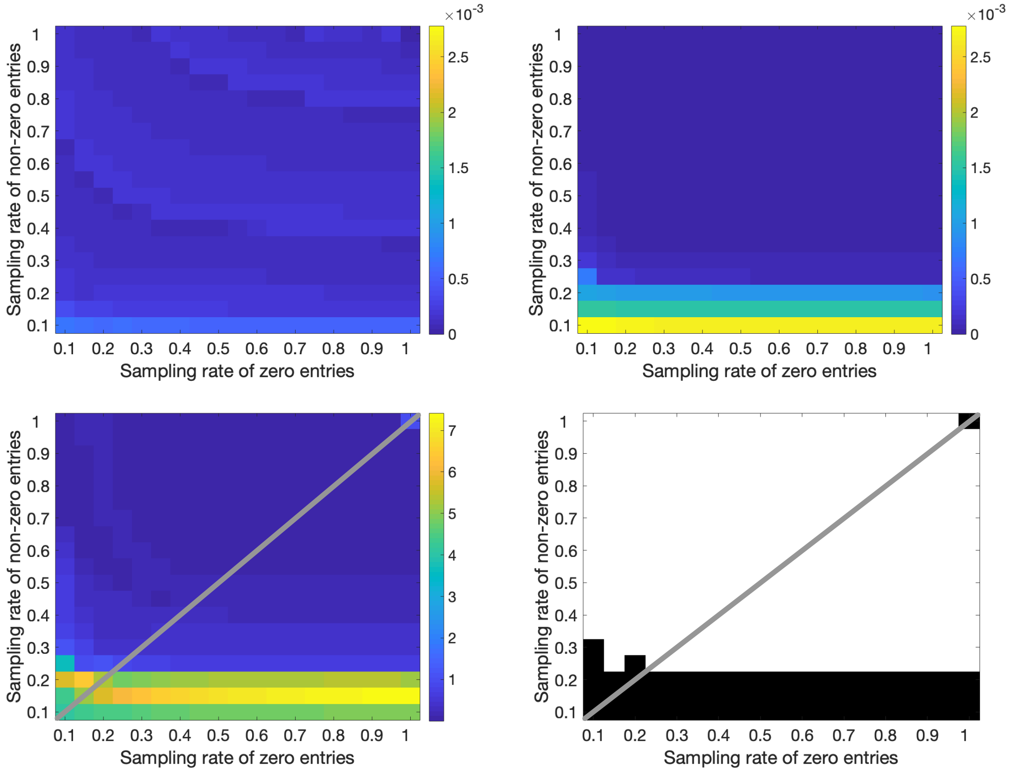

Figure 2.1: We consider twenty 1000×1000 sparse random matrices of rank 10 with average density equal to 0.66. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{M}-\tilde{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\tilde{\mathbf{X}}\|_F$, and (right) the binned average ratio where white means the average ratio is strictly less than 1, and black otherwise.

500×500 **rank 10 matrices**

In Figure 2.2, we construct twenty sparse random matrices of size 500×500 and of rank 10, as described in Section 2.4.2. We observe that Structured sIRLS outperforms sIRLS not only in the majority of the structured cases, but also in many of the other cases where the submatrix of unobserved entries is not necessarily sparse.

100×100 **rank 10 matrices**

In Figure 2.3, we construct twenty random matrices of size 100×100 and of rank 10, as described in Section 2.4.2. We observe in Figure 2.3 that Structured sIRLS outperforms sIRLS when the sampling rate of the nonzero entries is high (roughly speaking, when the decimal percent-

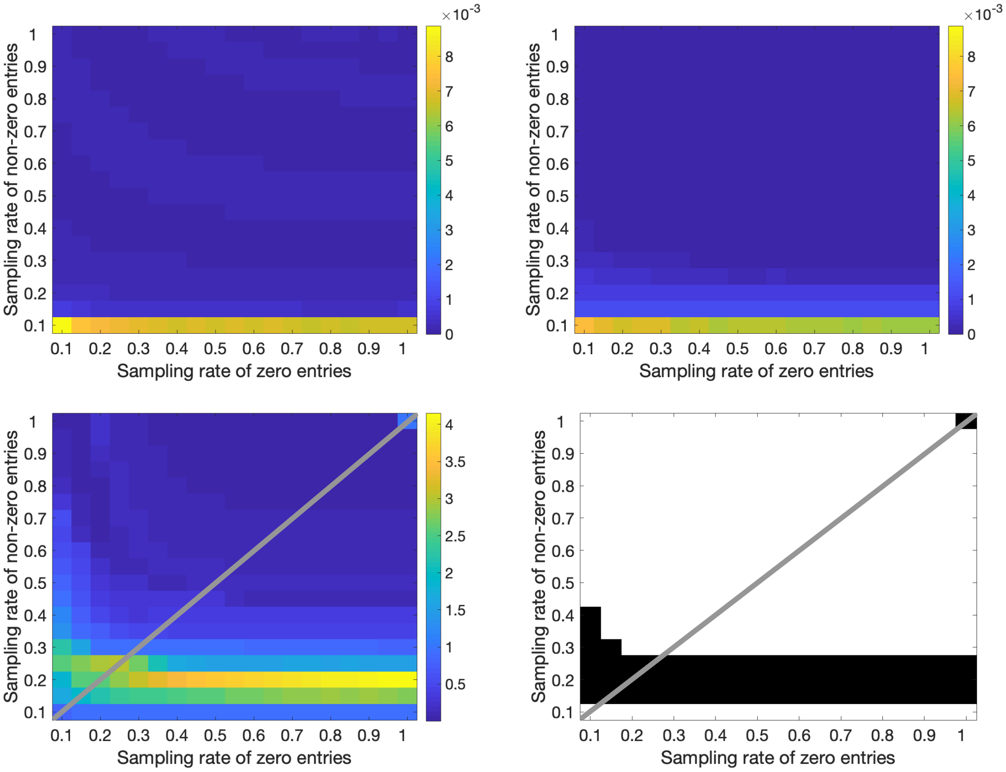

Figure 2.2: We consider twenty 500 × 500 sparse random matrices of rank 10 with average density equal to 0.66. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{M}-\tilde{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\tilde{\mathbf{X}}\|_F$, and (right) the binned average ratio where white means the average ratio is strictly less than 1, and black otherwise.

age of sampled nonzero entries is greater than 0.5), which covers the majority of the cases where there is sparsity-based structure in the missing entries.

100 × 100 **matrices with no knowledge of the rank a priori**

In Figure 2.4, we construct twenty random matrices of size 100 × 100 and of rank 8, as described in Section 2.3.3. For this experiment, we do not provide the algorithm with any rank estimate, for either sIRLS or Structured sIRLS. Instead, we allow the algorithm to estimate the rank at each iteration based on a heuristic described in Section 2.3.3. We observe in the bottom right plot of Figure 2.4, where we zoom in on the cases where the sampling rate of non-zero entries is at least 0.7, that Structured sIRLS outperform sIRLS to some extent in this region. Indeed, Structured sIRLS does particularly better when more entries are observed. As a reminder, the

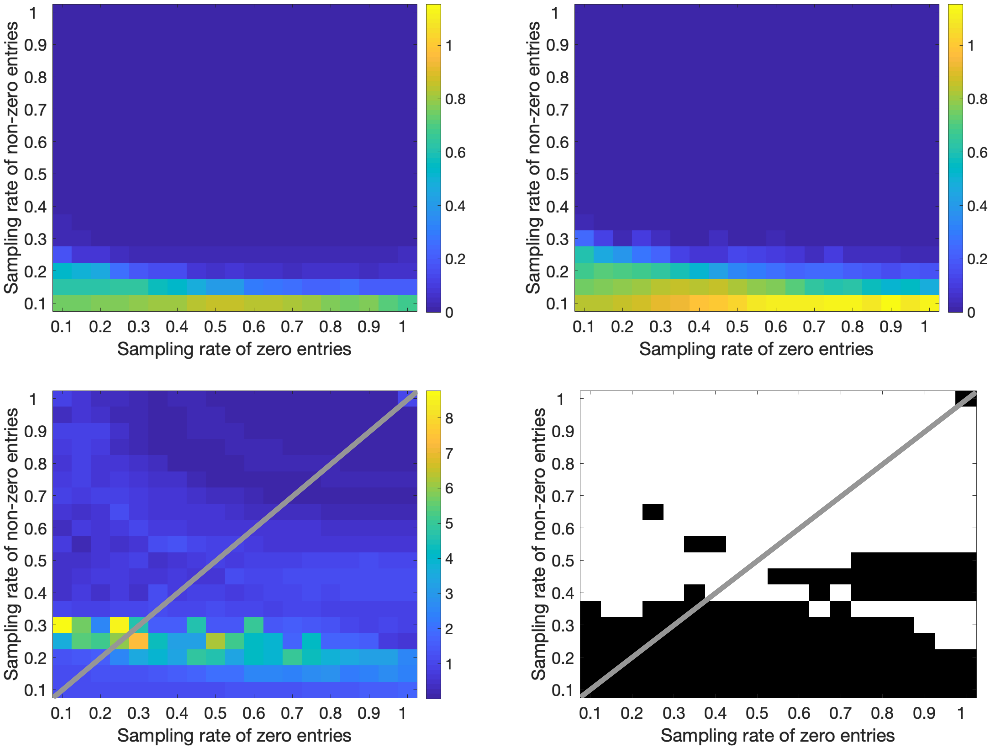

Figure 2.3: We consider twenty 100 × 100 sparse random matrices of rank 10 with average density equal to 0.66. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{M}-\tilde{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\tilde{\mathbf{X}}\|_F$, and (right) the binned average ratio where white means the average ratio is strictly less than 1, and black otherwise.

region above the diagonal gray lines is the region where the sampling rate of non-zero entries is greater than the sampling rate of zero entries.

100 × 100 **rank 20 matrices**

We say a matrix completion problem is hard when the degrees of freedom ratio FR is greater than 0.4 (as in [117]). In the previous experiments, we considered a few cases where $FR > 0.4$, which occur when the sampling rates of zero and nonzero entries are both relatively small. In these cases, there is not necessarily high sparsity-based structure, which imposes another challenge since the sampling rate of non-zero entries is approximately equal to or only slightly greater than the sampling rate of zero entries. Therefore, in this section, we consider hard cases (where $FR > 0.4$) with sparsity-based structure.

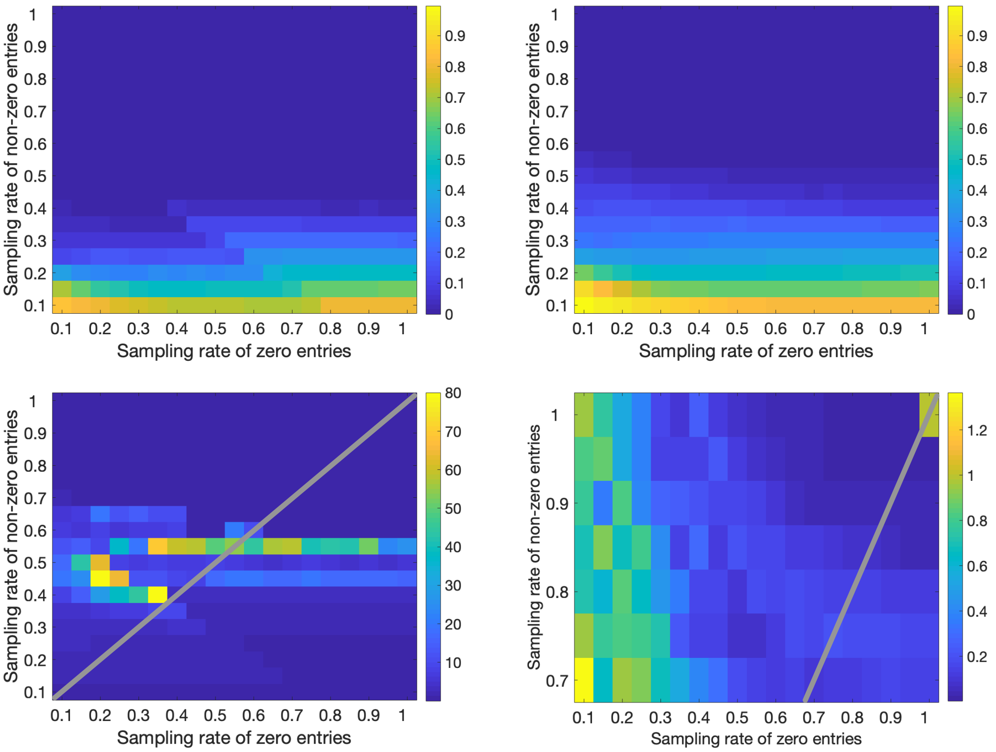

Figure 2.4: We consider twenty 100 × 100 sparse random matrices of rank 8 with density equal to 0.58, but we do not input any rank guess. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{M}-\tilde{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\tilde{\mathbf{X}}\|_F$, and (right) the average ratio when the sampling rate of non-zero entries is at least 0.70 (a zoomed in version of part of the lower left plot).

In Figure 2.5, we construct twenty random matrices of size 100 × 100 and of rank 20, as described in Section 2.4.2. If the number of sampled entries is 90% of the entire matrix, i.e. $|\Omega| = 0.9 \cdot m \cdot n$, then

$$FR = r(m+n-r)/|\Omega| = 20(200-20)/(0.9\times 100^2) = 0.4.$$

So, even sampling 90% of the matrix is still considered to be a hard problem. In the bottom row of Figure 2.5, the added red line separates the “hard" cases from those that are not: all the cases below the red line are hard. Note that in these hard regimes with sparsity-based structure, Structured sIRLS outperform sIRLS more often than not.

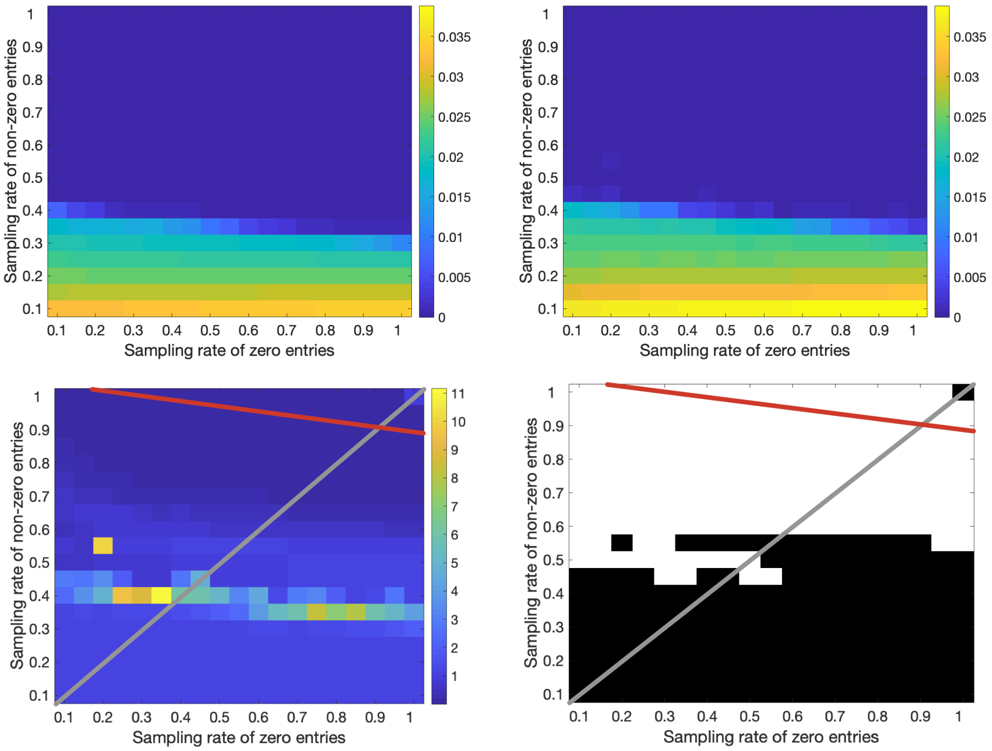

Figure 2.5: We consider twenty 100 × 100 sparse random matrices of rank 20 with average density equal to 0.88. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{M}-\tilde{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\tilde{\mathbf{X}}\|_F$, and (right) the binned average ratio where white means the average ratio is strictly less than 1, and black otherwise. The red line separates the hard cases from those that are not: all the cases below the red line are hard.

Comparison with Structured NNM on 30 × 30 rank 7 matrices

In this section, we run numerical experiments to compare the performance of Structured sIRLS with Structured NNM, using the L_1 norm on the submatrix of unobserved entries for Structured NNM. We use CVX, a package for specifying and solving convex programs [60, 61], to solve Structured NNM. In the experiments of Figure 2.6, we construct twenty random matrices of size 30 × 30 and of rank 7 as described in Section 2.4.2. We compare the accuracy with Structured NNM on small-sized matrices due to computational constraints of CVX: with this implementation of Structured NNM, it is difficult to handle significantly larger matrices. Similar experiments are considered in [118], where Structured NNM is compared to NNM. Comparing our iterative algorithm to Structured NNM is important since Structured NNM adapts nuclear norm minimization and ℓ_1 norm minimization, which are common heuristics for minimizing rank and inducing sparsity, respectively. We define the *relative error of Structured NNM* as $\|\mathbf{M}-\bar{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, where $\bar{\mathbf{X}}$ is the output of the Structured NNM algorithm. The average ratio is then defined as $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\bar{\mathbf{X}}\|_F$, where $\hat{\mathbf{X}}$ is the output of the Structured sIRLS algorithm.

For all sampling rates, the degrees of freedom ratio is greater than 0.4, i.e. all the problems are considered to be "hard" matrix completion problems. In Figure 2.6, we provide Structured sIRLS with the rank of the matrices.

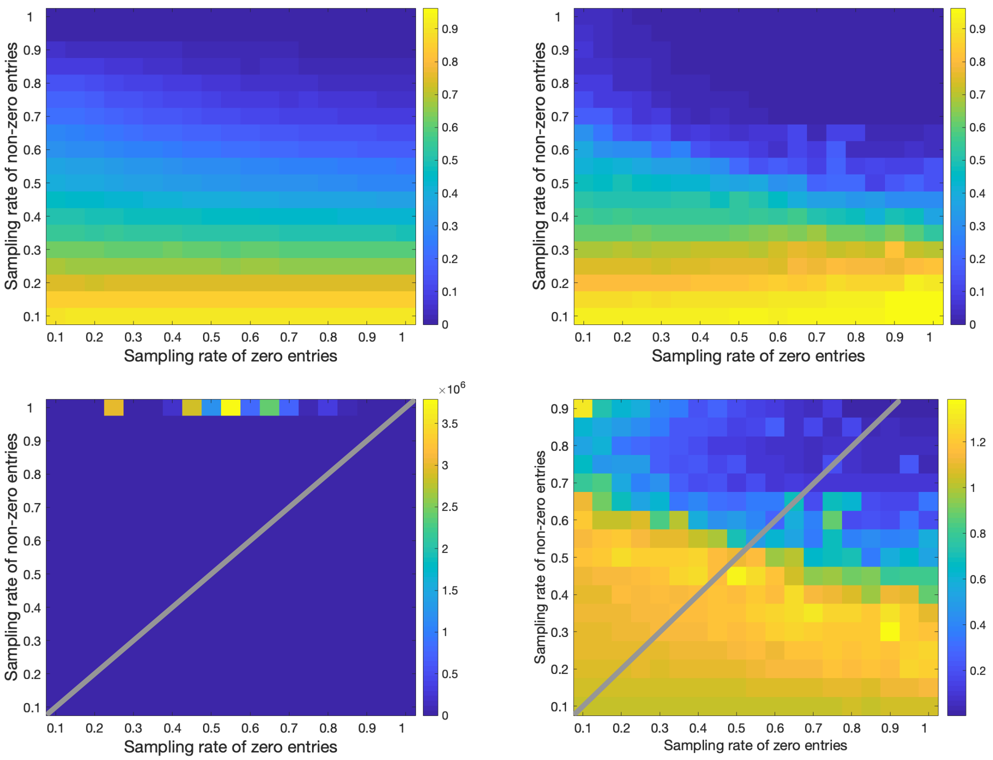

Figure 2.6: We consider twenty 30 × 30 sparse random matrices of rank 7, with average density equal to 0.53. We provide Structured sIRLS with the rank of the matrices. We optimize for each matrix and combination of sampling rates the regularization parameter $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ for Structured NNM and report the "optimal" results. The upper plots display (left) the average relative error for Structured NNM $\|\mathbf{M}-\bar{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{M}-\hat{\mathbf{X}}\|_F/\|\mathbf{M}-\bar{\mathbf{X}}\|_F$, and (right) the average ratio when the sampling rate of non-zero entries is at most 0.90 (a zoomed in version of part of the lower left plot).

In Figure 2.6, we give Structured NNM an advantage by optimizing for each matrix and combination of sampling rates the regularization parameter $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ for Structured NNM. However, for Structured sIRLS (again with $p = q = 1$) we do not optimize the gradient step sizes or the number of step sizes. Varying the number or sizes of the gradient steps controls how much we would like to promote low-rankness versus sparsity in the submatrix of missing entries. In the experiments of Figure 2.6, we observe that for the most part where the sampling rate of nonzero entries is between 0.6 and 0.9, Structured sIRLS performs better than Structured NNM. Furthermore, for the remainder of the structured settings, Structured sIRLS performs approximately the same as Structured NNM or only slightly worse. We note that in a couple of cases where the sampling rate of nonzero entries is 1, and where the relative error for both algorithms is close to zero, Structured NNM performs much better. This is in part because we optimize Structured NNM over $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$; see [118] where the relationship between the choice of α and the accuracy of Structured NNM is studied.

As observed in Figure 2.6, our proposed method is often comparable to Structured NNM on small-sized matrices, with certain regions where Structured sIRLS results in improved accuracy. In addition, iterative methods are well-known to offer ease of implementation and reduced computational resources, making our approach attractive not only in the setting of small-sized matrices, but also large-sized matrices.

2.4.3 Matrix Completion with Noise

In this section, we investigate the performance of Structured sIRLS when the observed entries are corrupted with noise. In particular, we compare the performance of Structured sIRLS with the performance of sIRLS. We adapt sIRLS and Structured sIRLS for noisy matrix completion by replacing the observed entries $P_\Omega(\mathbf{M})$ with the noisily observed entries $P_\Omega(\mathbf{B})$ in the constraints, where M is an unknown low-rank matrix that we wish to recover, where $P_\Omega(\mathbf{Z})$ is the measurement noise, and where the noisy matrix $\mathbf{B}$ satisfies $P_\Omega(\mathbf{B}) = P_\Omega(\mathbf{M}) + P_\Omega(\mathbf{Z})$. The algorithms for matrix recovery do not update the noisily observed entries, only the missing entries. We define our noise model such that $\|P_\Omega(\mathbf{Z})\|_F = \epsilon \|P_\Omega(\mathbf{M})\|_F$ for a noise parameter ϵ. We do so by adding noise of the form

$$\mathbf{Z}_{ij} = \epsilon \cdot \frac{\|P_\Omega(\mathbf{M})\|_F}{\|P_\Omega(\mathbf{N})\|_F} \cdot \mathbf{N}_{ij},$$

where $\mathbf{N}_{ij}$ are i.i.d. Gaussian random variables with the standard distribution $\mathcal{N}(0,1)$. We define the relative error of Structured sIRLS as $\|\mathbf{B}-\hat{\mathbf{X}}\|_F/\|\mathbf{B}\|_F$, where $\hat{\mathbf{X}}$ is the output of the Structured sIRLS algorithm. Similarly, we define the relative error of sIRLS as $\|\mathbf{B}-\tilde{\mathbf{X}}\|_F/\|\mathbf{B}\|_F$, where $\tilde{\mathbf{X}}$ is the output of the sIRLS algorithm.

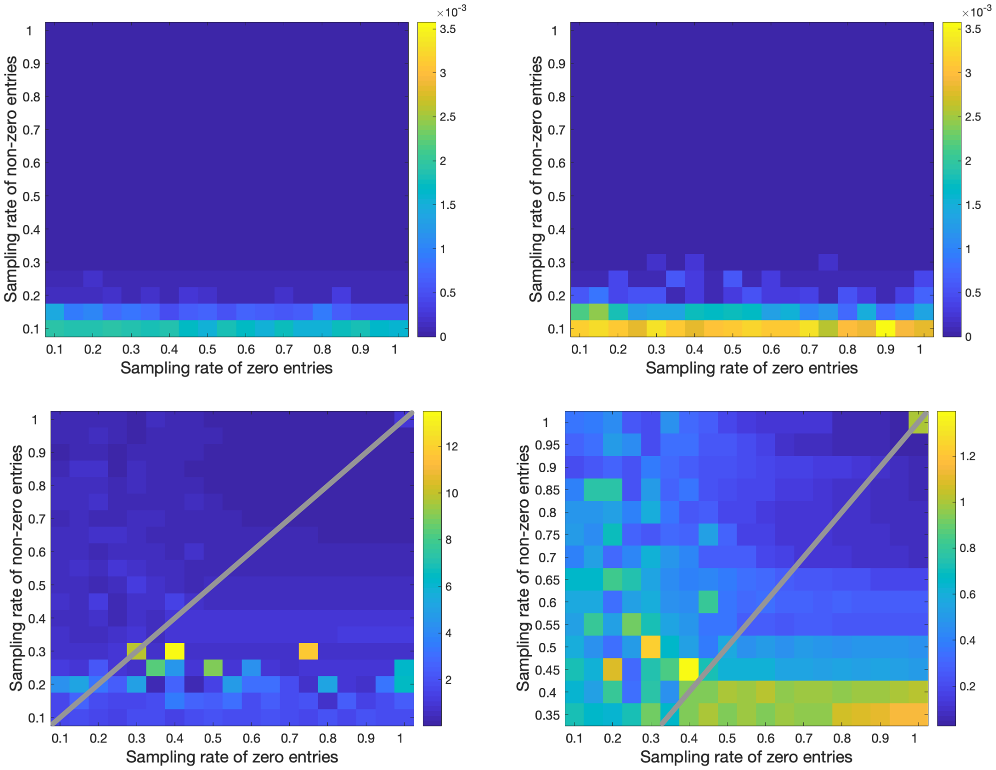

Figure 2.7: We consider twenty 100×100 random matrices of rank 3 with noise parameter $\epsilon = 10^{-4}$. The upper plots display (left) the average relative error for sIRLS $\|\mathbf{B}-\tilde{\mathbf{X}}\|_F/\|\mathbf{B}\|_F$, and (right) the average relative error for Structured sIRLS $\|\mathbf{B}-\hat{\mathbf{X}}\|_F/\|\mathbf{B}\|_F$. The lower plots display (left) the average ratio $\|\mathbf{B}-\hat{\mathbf{X}}\|_F/\|\mathbf{B}-\tilde{\mathbf{X}}\|_F$, and (right) the average ratio when the sampling rate of non-zero entries is at least 0.35 (a zoomed in version of part of the lower left plot).

In Figure 2.7, we consider twenty random 100×100 rank 3 matrices with noise parameter $\epsilon = 10^{-4}$, where we construct our matrices in the same fashion as in Section 2.4.2. We consider analogous structured settings as in the prior experiments, and observe that for the cases where the sampling rate of nonzero entries is greater than 0.3, which covers the majority of the cases where there is sparsity-based structure in the missing entries, Structured sIRLS performs better than sIRLS. For a higher noise level $\epsilon = 10^{-3}$, we observe that sIRLS and Structured sIRLS algorithms perform roughly the same. This suggest that both sIRLS and Structured sIRLS are robust to noise, with the improvements of Structured sIRLS from the structure diminishing as the noise grows.

2.5 Conclusion

In this work, we consider the notion of structured matrix completion, studied in the recent paper [118]. In particular, we are interested in sparsity-based structure in the missing entries whereby the vector of missing entries is close in the ℓ_0 or ℓ_1 norm sense to the zero vector (or more generally, to a constant vector). For example, a missing rating of a movie might indicate the user's lack of interest in that movie, thus suggesting a lower rating than otherwise expected. In [118], Molitor and Needell propose adjusting the standard nuclear norm minimization problem by regularizing the values of the unobserved entries to take into account the structural differences between observed and unobserved entries.

To our knowledge, we develop the first iterative algorithm that addresses the structured low-rank matrix completion problem, for which Structured NNM has been proposed. We adapt an iterative algorithm, called Structured IRLS, by adjusting the IRLS algorithm proposed in [117]. We also present a gradient-projection-based implementation, called Structured sIRLS, that can handle large-scale matrices. The algorithms are designed to promote low-rank structure in the recovered matrix with sparsity in the missing entries.

We perform numerical experiments on various structured settings to test the performance Structured sIRLS compared to sIRLS and Structured NNM. We consider problems of various degrees of freedom and rank knowledge. To generate matrices with sparsity-based structure in the missing entries, we subsample from the zero and nonzero entries of a sparse data matrix at various rates. We are particularly interested in the structured cases, i.e. the cases where a missing entry is more likely to be zero. This occurs when the sampling rate of the zero entries is lower than the sampling rate of the nonzero entries.

Our numerical experiments show that Structured sIRLS often gives better recovery results than sIRLS in structured settings. Further, for small enough noise our proposed method often performs better than sIRLS in structured settings, and as noise gets larger both converge to the same performance. Further, our numerical experiments show that Structured sIRLS is comparable to Structured NNM on small-sized matrices, with Structured sIRLS performing better in various structured regimes.

In future work, we hope to extend the theoretical results for Structured IRLS to more general settings. In the simplified setting, in which all of the unobserved entries are exactly zero, we show that the approximation given by an iteration of Structured IRLS will always perform at least as well as that of IRLS with the same weights assigned. However, we empirically observe the stronger result that Structured sIRLS often outperforms sIRLS in structured settings (in which algorithms are run until convergence, and in which not all missing entries are zero). Another extension is to explore Structured IRLS for different values of p and q, both empirically and theoretically. Furthermore, a possible direction for future work is to extend sparsity-based structure in the missing entries to a more general notion of structure, whereby the probability that an entry is observed or not may depend on more than just the value of that entry. For example, one could imagine that columns in a matrix corresponding to popular movies would have many entries (user ratings) filled in. In this context, an entry might be more likely to be observed if many entries in its same column are also observed.

Chapter 3

Semi-supervised NMF Models for Topic Modeling in Learning Tasks

3.1 Introduction

Frequently, one is faced with the problem of performing a (semi-)supervised learning task on extremely high-dimensional data which contains redundant information. A common approach is to first apply a dimensionality-reduction or feature extraction technique (e.g., Principal Component Analysis [130]), and then train the model for the learning task on the new, learned representation of the data. One problematic aspect of this two-step approach is that the learned representation of the data may provide “good" fit, but could suppress data features which are integral to the learning task [67]. For this reason, supervision-aware dimensionality-reduction models have become increasingly important in data analysis; such models aim to use supervision in the process of learning the lower-dimensional representation, or even learn this representation alongside the supervised learning model [17, 133, 164].

A popular technique for *topic modeling* that provides a lower rank approximation of a matrix is nonnegative matrix factorization (NMF). Given a nonnegative matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$ and a target dimension $r \in \mathbb{N}$, NMF decomposes $\mathbf{X}$ into a product of two low-dimensional nonnegative matrices. The model seeks $\mathbf{A}$ and $\mathbf{S}$ so that $\mathbf{X} \approx \mathbf{AS}$, where $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$ is called the dictionary matrix and $\mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$ is called the representation matrix. Typically, r is chosen such that $r < \min\{n_1, n_2\}$ to reduce the dimension of the original data matrix or reveal latent themes in the data. Data points are typically stored as columns of $\mathbf{X}$, thus n_1 represents the number of features, and n_2 represents the number of samples. The columns of $\mathbf{A}$ are generally referred to as *topics*, which are characterized by features of the dataset. Each column of $\mathbf{S}$ provides the approximate representation of the respective column in $\mathbf{X}$ in the lower-dimensional space spanned by the columns of $\mathbf{A}$. Thus, the data points are well approximated by an additive linear combination of the latent topics.

Several formulations for this nonnegative approximation, $\mathbf{X} \approx \mathbf{AS}$, have been studied [41, 102, 103, 171]; e.g.,

$$\operatorname*{argmin}_{\mathbf{A}\geq 0, \mathbf{S}\geq 0} \|\mathbf{X} - \mathbf{AS}\|_F^2 \quad \text{and} \quad \operatorname*{argmin}_{\mathbf{A}\geq 0, \mathbf{S}\geq 0} D(\mathbf{X}\|\mathbf{AS}), \tag{3.1}$$

where $\mathbf{A} \geq 0$ denotes a matrix restricted to have only nonnegative entries, $\|\cdot\|_F$ is the Frobenius norm, and $D(\cdot\|\cdot)$ is the information divergence that we define in (3.5). In what follows, we refer to the left formulation of (3.1) as $\|\cdot\|_F$-NMF and the right formulation of (3.1) as $D(\cdot\|\cdot)$-NMF. We refer the reader to [41] for discussions of similarity measures and generalized divergences (where information divergence is a particular case), and [106, 155] for generalized nonnegative matrix approximations with Bregman divergences.

Multiplicative update algorithms for both formulations of (3.1) have been proposed [102, 103]. These algorithms are widely adopted because they are easy to implement, do not require user-specified hyperparameters, preserve the nonnegativity constraints, and have desirable monotonicity properties [102]. Other popular algorithms include projected gradient descent and alternating least-squares [41, 91, 92, 107].

NMF has gained popularity recently due to large scale data demands of applications such as document clustering [12, 57, 127, 149, 168], image processing [66, 79, 103], financial data mining [44], audio processing [40, 58], and genetics [109]. Nonegativity of the factor matrices allows for better interpretabilty in many applications where the features are naturally nonnegative (e.g. pixel values, word counts, etc.). The classic work of [103] demonstrates that multiplicative updates of NMF is able to learn parts-based, sparse representation of the data (e.g. parts of faces and semantic features of text). This is in contrast to other methods, such as principal components analysis and vector quantization, that learn holistic, not parts-based, representations.

Semi-supervised Nonnegative Matrix Factorization (SSNMF), proposed in [104], is a modification of NMF to jointly incorporate a data matrix and a (partial) class label matrix. Following [104], let $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$ denote the data matrix and $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{k \times n_2}$ the supervision matrix, where the

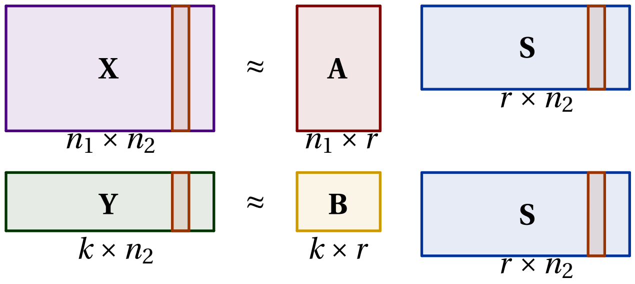

Figure 3.1: Given the number of classes k, and a desired dimension r, SSNMF is formulated as a joint factorization of a data matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$ and a label matrix $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{k \times n_2}$, sharing representation factor $\mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$.

data observations are the columns of $\mathbf{X}$ and the associated targets (e.g., labels) are the columns of $\mathbf{Y}$. SSNMF seeks $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$, $\mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$, and $\mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$ which jointly factorize $\mathbf{X}$ and $\mathbf{Y}$; that is $\mathbf{X} \approx \mathbf{AS}$ and $\mathbf{Y} \approx \mathbf{BS}$. SSNMF is defined by

$$\underset{\mathbf{A},\mathbf{S},\mathbf{B} \geq 0}{\operatorname{argmin}} \underbrace{\|\mathbf{W} \odot (\mathbf{X} - \mathbf{AS})\|_F^2}_{\text{Reconstruction Error}} + \lambda \underbrace{\|\mathbf{L} \odot (\mathbf{Y} - \mathbf{BS})\|_F^2}_{\text{Classification Error}}, \tag{3.2}$$

where $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$, $\mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$, $\mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$, and the regularization parameter $\lambda > 0$ governs the relative importance of the supervision term [104]. See Figure 3.1 for an illustration of the SSNMF model. We denote this objective function as $F_1(\mathbf{A}, \mathbf{B}, \mathbf{S}; \mathbf{X}, \mathbf{Y})$. The binary weight matrix $\mathbf{W}$ accommodates missing data by indicating observed and unobserved data entries (that is, $\mathbf{W}_{ij} = 1$ if $\mathbf{X}_{ij}$ is observed, and $\mathbf{W}_{ij} = 0$ otherwise). Similarly, $\mathbf{L} \in \mathbb{R}^{k \times n_2}$ is a weight matrix that indicates the presence or absence of a label (that is, $\mathbf{L}_{\bullet,j} = \mathbf{1_k}$ if the label of $\mathbf{X}_{\bullet,j}$ is known, and $\mathbf{L}_{\bullet,j} = \mathbf{0_k}$ otherwise).

Multiplicative updates have been developed for SSNMF for the Frobenius norm, and the resulting performance of clustering and classification is improved by incorporating data labels into NMF [104]. To differentiate the model defined in (3.2) from the proposed SSNMF models, we refer to the model defined by (3.2) as $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF.

In this work, we define models with different error functions applied to the reconstruction and supervision factorization terms as

$$\underset{\mathbf{A},\mathbf{S},\mathbf{B}\geq 0}{\operatorname{argmin}} \underbrace{R(\mathbf{W}\odot\mathbf{X},\mathbf{W}\odot\mathbf{AS})}_{\text{Reconstruction Error}} + \lambda \underbrace{S(\mathbf{L}\odot\mathbf{Y},\mathbf{L}\odot\mathbf{BS})}_{\text{Supervision Error}} \tag{3.3}$$

denoted by $(R(\cdot,\cdot), S(\cdot,\cdot))$-SSNMF where $R(\cdot,\cdot)$ and $S(\cdot,\cdot)$ are the error functions applied to the reconstruction term and supervision term, respectively. In each model, the matrix $\mathbf{A}\in\mathbb{R}_{\geq 0}^{n_1\times r}$ provides a basis for the lower-dimensional space, $\mathbf{S}\in\mathbb{R}_{\geq 0}^{r\times n_2}$ provides the coefficients representing the projected data in this space, and $\mathbf{B}\in\mathbb{R}_{\geq 0}^{k\times r}$ provides the supervision model which predicts the targets given the representation of points in the lower-dimensional space. We allow for missing data and labels or confidence-weighted errors via the data-weighting matrix $\mathbf{W}\in\mathbb{R}_{\geq 0}^{n_1\times n_2}$ and the label-weighting matrix $\mathbf{L}\in\mathbb{R}_{\geq 0}^{k\times n_2}$.

We point out the simple fact that these joint factorizations can be stacked into a single NMF (visualized in Figure 3.1)

$$\begin{bmatrix}\mathbf{X}\\ \mathbf{Y}\end{bmatrix} \approx \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\mathbf{S}. \tag{3.4}$$

which will be useful in some of the analysis in the next sections.

Table 3.1: Overview of NMF and SSNMF models.

Model	Objective
$\|\cdot\|_F$-NMF [103]	$\underset{\mathbf{A},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{AS}\|_F^2$
$D(\cdot\|\cdot)$-NMF [102]	$\underset{\mathbf{A},\mathbf{S}\geq 0}{\operatorname{argmin}} D(\mathbf{X}\|\mathbf{AS})$
$(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF [104]	$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{W}\odot(\mathbf{X}-\mathbf{AS})\|_F^2 + \lambda\|\mathbf{L}\odot(\mathbf{Y}-\mathbf{BS})\|_F^2$
$(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF	$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{W}\odot(\mathbf{X}-\mathbf{AS})\|_F^2 + \lambda D(\mathbf{L}\odot\mathbf{Y}\|\mathbf{L}\odot\mathbf{BS})$
$(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF	$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} D(\mathbf{W}\odot\mathbf{X}\|\mathbf{W}\odot\mathbf{AS}) + \lambda\|\mathbf{L}\odot(\mathbf{Y}-\mathbf{BS})\|_F^2$
$(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF	$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} D(\mathbf{W}\odot\mathbf{X}\|\mathbf{W}\odot\mathbf{AS}) + \lambda D(\mathbf{L}\odot\mathbf{Y}\|\mathbf{L}\odot\mathbf{BS})$

In Table 3.1, we summarize existing and proposed models, where each proposed model is of the form (3.3) for specific choices of error functions R and S. Our models differ from that of [104] in the error functions used, since our models utilize information divergence on the data reconstruction term. This is a natural choice since many representations of document data (e.g., bag-of-words, n-grams, etc.) correspond to counts of word patterns in the data and are naturally modelled by Poisson distribution(s), which leads to the information divergence in the maximum likelihood estimation (MLE) model [37, 74, 121, 122, 135, 143]. Furthermore, our proposed models differ in the classification framework proposed (in Section 3.2.4) which does not necessarily rely on support vector machines (SVM) for linear classification. We further provide analysis on the topics learned for the classification task where the choice of rank is not necessarily the same as the number of classes.

3.1.1 Background and Related Work

In this section, we describe related work most relevant to our own. We focus on work in three main areas: statistical motivation for NMF models, models for simultaneous dimension reduction and supervised learning, and semi-supervised and joint NMF models.

Statistical Motivation for NMF

The most common discrepancy measures for NMF are the Frobenius norm and the information divergence. One reason for this popularity is that $\|\cdot\|_F$-NMF and $D(\cdot\|\cdot)$-NMF correspond to the MLE given an assumed latent generative model and a Gaussian and Poisson model of uncertainty, respectively [28, 48, 160]. In [28, 160], the authors go further towards a Bayesian approach, introduce application-appropriate prior distributions on the latent factors, and apply *maximum a posteriori* (MAP) estimation. Under certain conditions, $D(\cdot\|\cdot)$-NMF is equivalent to probabilistic latent semantic indexing [45].

Dimension Reduction and Learning

There has been much work developing dimensionality-reduction models that are supervision-aware. Semi-supervised clustering makes use of known label information or other supervision *and* the data features while forming clusters [6, 93, 162]. These techniques generally make use of label information in the cluster initialization or during cluster updating via must-link and cannot-link constraints; empirically, these approaches are seen to increase mutual information between computed clusters and user-assigned labels [6]. Semi-supervised feature extraction makes use of supervision information in the feature extraction process [56,151]. These approaches are generally *filter*- or *wrapper*-based approaches, and distinguished by their underlying supervision type [151].

Linear Discriminant Analysis [53, 136] is another popular linear dimensionality technique used in supervised learning, e.g. as a linear classifier or a pre-processing step before classification. Principal Component Analysis (PCA) is an unsupervised technique that searches for directions in the data that have the largest variance. Linear Discriminant Analysis is supervised technique that searches for directions that maximize class separation (i.e. a large variance among the classes, and a small variance within each class).

Semi-supervised and Joint NMF

Since the seminal work of Lee et al. [104], semi-supervised NMF models have been studied in a variety of settings. The works [33,51,84] propose models which exploit cannot-link or must-link supervision. In [38], the authors introduce a model with information divergence penalties on the reconstruction and on supervision terms which influence the learned factorization to approximately reconstruct coefficients learned before factorization by a support-vector machine (SVM). Several works [85, 169, 173] propose a supervised NMF model that incorporates Fisher discriminant constraints into NMF for classification. Furthermore, joint factorization of two data matrices, like that of SSNMF, is described more generally and denoted Simultaneous NMF in [41].

3.1.2 Contribution

In this work, we propose variants of semi-supervised nonnegative matrix factorization formulations which utilize information divergence on the data reconstruction term. The Poisson distribution is particularly well-suited for integer-valued datasets, such a documents represented by vector of term frequencies [135], or images which can be interpreted as a photon counting process [72], which leads to the information divergence in the MLE model [37,74,121]. As in [104], our proposed models generalize NMF to supervised learning tasks and provide a topic model which simultaneously provides a lower dimensional representation of the data and a predictive model for targets. While historically the focus of SSNMF studies have been on classification [104], we highlight that this joint factorization model can be applied quite naturally to regression tasks. The main contributions of this work are as follows:

- **Maximum likelihood estimators.** We motivate the proposed SSNMF models and that of [104] as maximum likelihood estimators (MLE) given specific models of uncertainty in the observations.

- **Multiplicative updates**. We derive multiplicative updates for the proposed models that allow for missing data and partial supervision.

- **Classification framework.** We propose a classification framework using SSNMF based on the linear classifier $\mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$ obtained from the factorization.

- **Experiments on the 20 Newsgroups dataset.** We perform experiments on the 20 Newsgroups benchmark dataset [141], illustrating the promise of these models in both topic modeling and classification relative to the performance of other common document classifiers (e.g. Linear SVM, Multinomial Naive Bayes).

Our PyPI package for all SSNMF models [68] and our code for reproducing experiments[5] are publicly available.

3.1.3 Organization

The chapter is organized as follows. We begin by motivating the proposed models and that of [104] via MLE in Section 3.2.2, present the multiplicative update methods for training in Section 3.2.3, and present details of a framework for classification with these models in Section 3.2.4. We present experimental evidence illustrating the promise of the SSNMF models on

[5] https://github.com/jamiehadd/ssnmf

the 20 Newsgroup dataset in Section 3.3. Finally, we end with some conclusions and discussion of future work in Section 3.4.

3.2 SSNMF Models: Motivation and Methods

In this section, we present a statistical MLE motivation for variants of the SSNMF model, introduce the general semi-supervised models, and provide a multiplicative updates method for each variant.

3.2.1 Notation

Our models make use of two matrix similarity measures. The first is the standard Frobenius norm, $\|\mathbf{A}-\mathbf{B}\|_F$. The second is the *information divergence* or I-divergence, a measure defined between nonnegative matrices $\mathbf{A}$ and $\mathbf{B}$,

$$D(\mathbf{A}\|\mathbf{B}) = \sum_{i,j}\left(\mathbf{A}_{ij}\log\frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}} - \mathbf{A}_{ij} + \mathbf{B}_{ij}\right), \tag{3.5}$$

where $D(\mathbf{A}\|\mathbf{B}) \geq 0$ with equality if and only if $\mathbf{A} = \mathbf{B}$ [102]. Because the information divergence reduces to the Kullback-Leibler divergence when $\mathbf{A}$ and $\mathbf{B}$ represent probability distributions, i.e., $\sum\mathbf{A}_{ij} = \sum\mathbf{B}_{ij} = 1$, it is often referred to as the generalized Kullback-Leibler divergence [48].

In the following, $\mathbf{A}/\mathbf{B}$ indicates element-wise division, $\mathbf{A}\odot\mathbf{B}$ indicates element-wise multiplication, and $\mathbf{AB}$ denotes standard matrix multiplication. We denote the set of non-zero indices of a matrix by $\text{supp}(\mathbf{A}) := \left\{(i,j) : \mathbf{A}_{ij} \neq 0\right\}$. When an $n_1 \times n_2$ matrix is to be restricted to have only nonnegative entries, we write $\mathbf{A} \geq 0$ and $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1\times n_2}$. We let $\mathbf{1_k}$ denote the length-k vector consisting of ones, $\mathbf{1_k} = (1,\cdots 1)^\top \in \mathbb{R}^k$, and similarly $\mathbf{0_k}$ denotes the vector of all zeros, $\mathbf{0_k} = (0,\cdots 0)^\top \in \mathbb{R}^k$.

We let $\mathcal{N}\left(z\middle|\mu,\sigma^2\right)$ denote the Gaussian density function for a random variable z with mean μ and variance σ^2, and $\mathscr{PO}\left(z|v\right)$ denotes the Poisson density function for a random variable z with nonnegative intensity parameter v.

3.2.2 Maximum Likelihood Estimation

In this section, we demonstrate that specific forms of our proposed variants of SSNMF are maximum likelihood estimators for given models of uncertainty or noise in the data matrices **X** and **Y**. These different uncertainty models have their likelihood function maximized by different error functions chosen for reconstruction and supervision errors, R and S. We summarize these results next; each MLE derived is a specific instance of a general model discussed in Section 3.2.3 or in [104].

Maximum Likelihood Estimators. Suppose that the observed data **X** and supervision information **Y** have entries given as the sum of random variables,

$$\mathbf{X}_{\gamma,\tau} = \sum_{i=1}^{r} x_{\gamma,i,\tau} \text{ and } \mathbf{Y}_{\eta,\tau} = \sum_{i=1}^{r} y_{\eta,i,\tau},$$

and that the set of $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$ are statistically independent conditional on $\mathbf{A}, \mathbf{B}$, and $\mathbf{S}$.

1. When $x_{\gamma,i,\tau}$ and $y_{\eta,i,\tau}$ have distributions

   $$\mathcal{N}\left(x_{\gamma,i,\tau}\middle|\mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau},\sigma_1\right) \text{ and } \mathcal{N}\left(y_{\eta,i,\tau}\middle|\mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau},\sigma_2\right)$$

   respectively, the maximum likelihood estimator is

   $$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{AS}\|_F^2 + \frac{\sigma_1}{\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2.$$

2. When $x_{\gamma,i,\tau}$ and $y_{\eta,i,\tau}$ have distributions

   $$\mathcal{N}\left(x_{\gamma,i,\tau}\middle|\mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau},\sigma_1\right) \text{ and } \mathcal{PO}\left(y_{\eta,i,\tau}\middle|\mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right)$$

   respectively, the maximum likelihood estimator is

   $$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{AS}\|_F^2 + 2r\sigma_1 D(\mathbf{Y}\|\mathbf{BS}).$$

3. When $x_{\gamma,i,\tau}$ and $y_{\eta,i,\tau}$ have distributions

$$\mathscr{PO}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right) \text{ and } \mathscr{N}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, \sigma_2\right)$$

respectively, the maximum likelihood estimator is

$$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} D(\mathbf{X}\|\mathbf{AS}) + \frac{1}{2r\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2.$$

4. When $x_{\gamma,i,\tau}$ and $y_{\eta,i,\tau}$ have distributions

$$\mathscr{PO}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right) \text{ and } \mathscr{PO}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right)$$

respectively, the maximum likelihood estimator is

$$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} D(\mathbf{X}\|\mathbf{AS}) + D(\mathbf{Y}\|\mathbf{BS}).$$

We note that 4 follows from [28, 48, 160], but the others are distinct from previous MLE derivations due to the difference in the distributions assumed on data $\mathbf{X}$ and supervision $\mathbf{Y}$. Here, we provide only the MLE derivation for 2 as the other derivations are similar; these are included in the appendix for completeness. We demonstrate that the MLE, in the case that the uncertainty on $\mathbf{X}$ is Gaussian distributed and on $\mathbf{Y}$ is Poisson distributed, is a specific instance of the $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF model.

Our models for the distribution of observed entries of $\mathbf{X}$ and $\mathbf{Y}$ assume that the mean is given by $\mathbb{E}[\mathbf{X}] = \mathbf{AS}$ and $\mathbb{E}[\mathbf{Y}] = \mathbf{BS}$, and the uncertainty in the set of observations in $\mathbf{X}$ is governed by a Gaussian distribution while the set in $\mathbf{Y}$ is governed by a Poisson distribution. That is, we consider hierarchical models for $\mathbf{X}$ and $\mathbf{Y}$ where

$$\mathbf{X}_{\gamma,\tau} = \sum_{i=1}^{r} x_{\gamma,i,\tau} \text{ and } x_{\gamma,i,\tau} \sim \mathscr{N}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, \sigma_1\right),$$

$$\mathbf{Y}_{\eta,\tau} = \sum_{i=1}^{r} y_{\eta,i,\tau} \text{ and } y_{\eta,i,\tau} \sim \mathscr{PO}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right).$$

Note then that

$$\mathbf{X}_{\gamma,\tau} \sim \mathscr{N}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, r\sigma_1\right), \text{ and } \mathbf{Y}_{\eta,\tau} \sim \mathscr{PO}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right)$$

due to the summable property of Gaussian and Poisson random variables. We note that this assumes different distributions on the two collections of rows of the single NMF (3.4), with Gaussian and Poisson models of uncertainty.

Assuming that the set of $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$ are statistically independent conditional on **A**, **B**, and **S**, we have that the likelihood

$$p(\mathbf{X}, \mathbf{Y}|\mathbf{A}, \mathbf{B}, \mathbf{S}) = \prod_{\gamma,\tau} \mathscr{N}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, r\sigma_1\right) \prod_{\eta,\tau} \mathscr{PO}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right). \tag{3.6}$$

We apply the monotonic natural logarithmic function to the likelihood and ignore terms that are invariant with the factor matrices. This transforms the likelihood into a $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF objective while preserving the maximizer. That is, the log likelihood (excluding additive terms which are constant with respect to **A**, **B**, and **S**) is

$$\begin{aligned}\ln p(\mathbf{X}, \mathbf{Y}|\mathbf{A}, \mathbf{B}, \mathbf{S}) &=^{+} -\frac{1}{2r\sigma_1} \sum_{\gamma,\tau} \left(\mathbf{X}_{\gamma,\tau} - \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right)^2 - \sum_{\eta,\tau} \left[(\mathbf{BS})_{\eta,\tau} - \mathbf{Y}_{\eta,\tau} \log(\mathbf{BS})_{\eta,\tau} + \log\Gamma\left(\mathbf{Y}_{\eta,\tau} + 1\right)\right] \\ &=^{+} -\frac{1}{2r\sigma_1} \left[\|\mathbf{X} - \mathbf{AS}\|_F^2 + 2r\sigma_1 D(\mathbf{Y}\|\mathbf{BS})\right].\end{aligned}$$

Here, $=^{+}$ denotes suppression of additive terms that do not depend upon **A**, **B**, and **S**, and $\Gamma(\cdot)$ denotes the Gamma function: $\Gamma(n) = (n-1)!$ for $n \geq 1$. Thus, the maximum likelihood estimators for **A**, **B**, and **S** are given by

$$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{AS}\|_F^2 + 2r\sigma_1 D(\mathbf{Y}\|\mathbf{BS}).$$

We see that the MLE, in the case of Gaussian uncertainty on the observations in **X** and Poisson uncertainty on the observations in **Y**, is a specific instance of the $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF objective where the regularization parameter λ is a multiple of the variance of the Gaussian distribution. The other MLEs are derived similarly; see Appendix A.1.

An instance of each of the models in Table 3.1 are MLE for a given model of uncertainty in the observed data **X** and supervision **Y**. While this motivates our exploration of these models, we present them in more general context next and provide training methods for the general form.

3.2.3 General Models and Multiplicative Updates

In this section, we propose the general form of $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF, $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF, and $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF and present multiplicative updates methods for each model. These three models are novel forms of SSNMF, and besides their statistical motivation via MLE, we demonstrate their promise experimentally in Section 3.3. Recall that $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF is defined by (3.2) and multiplicative updates are derived in [104].

As in [104], our multiplicative updates methods allow for missing (or certainty-weighted) data and missing (or certainty-weighted) supervision information via matrices **W** and **L**, which represent our knowledge or certainty of the corresponding entries of **X** and **Y**, respectively. When **W** is a matrix of all ones (or more generally has all equal entries) and **L** is a matrix of all zeros, the SSNMF models reduce to either the $\|\cdot\|_F$-NMF or $D(\cdot\|\cdot)$-NMF. The SSNMF model is fully supervised when $\text{supp}(\mathbf{Y}) \subset \text{supp}(\mathbf{L})$ and **Y** contains supervision information for each element in **X**.

The multiplicative updates for all methods can be derived as follows. Suppose that the gradient of the objective function F with respect to one of the variables Θ has a decomposition that is of the form:

$$\nabla_\Theta F = [\nabla_\Theta F]^+ - [\nabla_\Theta F]^-,$$

where $[\nabla_\Theta F]^+ > 0$ and $[\nabla_\Theta F]^- > 0$. Then we define the multiplicative update for Θ as:

$$\Theta \leftarrow \Theta - \Gamma \odot \nabla_\Theta F = \Theta - \Gamma \odot ([\nabla_\Theta F]^+ - [\nabla_\Theta F]^-) = \Theta \odot \frac{[\nabla_\Theta F]^-}{[\nabla_\Theta F]^+}$$

for $\Gamma = \dfrac{\Theta}{[\nabla_\Theta F]^+}$. We now provide detailed derivations of the multiplicative updates for the three proposed methods.

The first proposed semi-supervised NMF model is $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF, which is defined by the solution to

$$\operatorname*{argmin}_{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{AS})\|_F^2 + \lambda D(\mathbf{L} \odot \mathbf{Y} \| \mathbf{L} \odot \mathbf{BS}). \tag{3.7}$$

We denote this objective function as $F_2(\mathbf{A}, \mathbf{B}, \mathbf{S}; \mathbf{X}, \mathbf{Y})$. Similar to the previous SSNMF model, this model seeks a joint factorization of the data matrix $\mathbf{X}$ and target matrix $\mathbf{Y}$; however, the error functions applied to the reconstruction and classification terms in the objective differ.

The multiplicative updates for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$ which minimize (3.7) are derived as follows. The gradient of the objective function of (3.7) with respect to $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{S}$ are, respectively,

$$\begin{aligned}
\nabla_{\mathbf{A}} F_2 &= -2[\mathbf{W} \odot (\mathbf{X} - \mathbf{AS})]\mathbf{S}^\top, \\
\nabla_{\mathbf{B}} F_2 &= \mathbf{LS}^\top - \left[\frac{\mathbf{L} \odot \mathbf{Y}}{\mathbf{L} \odot \mathbf{BS}} \odot \mathbf{L}\right] \mathbf{S}^\top, \text{ and} \\
\nabla_{\mathbf{S}} F_2 &= \lambda \mathbf{B}^\top \mathbf{L} - \lambda \mathbf{B}^\top \left[\frac{\mathbf{L} \odot \mathbf{Y}}{\mathbf{L} \odot \mathbf{BS}} \odot \mathbf{L}\right] - 2\mathbf{A}^\top [\mathbf{W} \odot (\mathbf{X} – \mathbf{AS})].
\end{aligned}$$

The multiplicative updates method, Algorithm 3, can be viewed as an entrywise gradient descent method, where the stepsizes are chosen individually for each entry of the updating matrix to ensure nonnegativity. That is, the updates in Algorithm 3 are given by

$$\begin{aligned}
\mathbf{A} &\to \mathbf{A} - \Gamma \odot \nabla_{\mathbf{A}} F_2 \quad \text{for} \quad \Gamma = \frac{\mathbf{A}}{2(\mathbf{W} \odot \mathbf{AS})\mathbf{S}^\top}, \\
\mathbf{B} &\to \mathbf{B} - \Gamma \odot \nabla_{\mathbf{B}} F_2 \quad \text{for} \quad \Gamma = \frac{\mathbf{B}}{\mathbf{LS}^\top}, \text{ and} \\
\mathbf{S} &\to \mathbf{S} - \Gamma \odot \nabla_{\mathbf{S}} F_2 \quad \text{for} \quad \Gamma = \frac{\mathbf{S}}{2\mathbf{A}^\top(\mathbf{W} \odot \mathbf{AS}) + \lambda \mathbf{B}^\top \mathbf{L}}.
\end{aligned}$$

Algorithm 3: $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF multiplicative updates

```
Require: $\mathbf{X}, \mathbf{W} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$, $\mathbf{Y}, \mathbf{L} \in \mathbb{R}_{\geq 0}^{k \times n_2}$, $r$, $\lambda$, $N$
Initialize $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$
for $i = 1, ..., N$ do
    $\mathbf{A} \leftarrow \mathbf{A} \odot \frac{(\mathbf{W} \odot \mathbf{X})\mathbf{S}^\top}{(\mathbf{W} \odot \mathbf{AS})\mathbf{S}^\top}$;
    $\mathbf{B} \leftarrow \frac{\mathbf{B}}{\mathbf{LS}^\top} \odot \left[ \frac{(\mathbf{L} \odot \mathbf{Y})}{(\mathbf{L} \odot \mathbf{BS})} \odot \mathbf{L} \right] \mathbf{S}^\top$;
    $\mathbf{S} \leftarrow \mathbf{S} \odot \frac{2\mathbf{A}^\top(\mathbf{W} \odot \mathbf{X}) + \lambda \mathbf{B}^\top \left[ \frac{(\mathbf{L} \odot \mathbf{Y})}{(\mathbf{L} \odot \mathbf{BS})} \odot \mathbf{L} \right]}{2\mathbf{A}^\top(\mathbf{W} \odot \mathbf{AS}) + \lambda \mathbf{B}^\top \mathbf{L}}$;
end for
```

The next proposed semi-supervised NMF model is $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF, defined by the solution to

$$\operatorname*{argmin}_{\mathbf{A},\mathbf{B},\mathbf{S} \geq 0} D(\mathbf{W} \odot \mathbf{X} \| \mathbf{W} \odot \mathbf{AS}) + \lambda \|\mathbf{L} \odot (\mathbf{Y} - \mathbf{BS})\|_F^2. \tag{3.8}$$

We denote this objective function as $F_3(\mathbf{A}, \mathbf{B}, \mathbf{S}; \mathbf{X}, \mathbf{Y})$. Again, this model seeks a joint factorization of the data matrix $\mathbf{X}$ and target matrix $\mathbf{Y}$; here the reconstruction error is penalized by the information divergence, while the supervision error is penalized by the Frobenius norm. Multiplicative updates for this model are provided in Algorithm 4. The multiplicative updates follow from those for $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF (Algorithm 3) by swapping the roles of $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{A}$ and $\mathbf{B}$.

Algorithm 4: $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF multiplicative updates

```
Require: $\mathbf{X}, \mathbf{W} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$, $\mathbf{Y}, \mathbf{L} \in \mathbb{R}_{\geq 0}^{k \times n_2}$, $r$, $\lambda$, $N$
Initialize $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$
for $i = 1, \ldots, N$ do
    $\mathbf{A} \leftarrow \frac{\mathbf{A}}{\mathbf{W}\mathbf{S}^\top} \odot \left[ \frac{(\mathbf{W} \odot \mathbf{X})}{(\mathbf{W} \odot \mathbf{AS})} \odot \mathbf{W} \right] \mathbf{S}^\top;$
    $\mathbf{B} \leftarrow \mathbf{B} \odot \frac{(\mathbf{L} \odot \mathbf{Y})\mathbf{S}^\top}{(\mathbf{L} \odot \mathbf{BS})\mathbf{S}^\top};$
    $\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{A}^\top \left[ \frac{(\mathbf{W} \odot \mathbf{X})}{(\mathbf{W} \odot \mathbf{AS})} \odot \mathbf{W} \right] + 2\lambda \mathbf{B}^\top (\mathbf{L} \odot \mathbf{Y})}{\mathbf{A}^\top \mathbf{W} + 2\lambda \mathbf{B}^\top (\mathbf{L} \odot \mathbf{BS})};$
end for
```

The third, and final, proposed semi-supervised NMF model is $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF, defined by the solution to

$$\underset{\mathbf{A},\mathbf{B},\mathbf{S} \geq 0}{\operatorname{argmin}} D(\mathbf{W} \odot \mathbf{X} \| \mathbf{W} \odot \mathbf{AS}) + \lambda D(\mathbf{L} \odot \mathbf{Y} \| \mathbf{L} \odot \mathbf{BS}). \tag{3.9}$$

We denote this objective function as $F_4(\mathbf{A}, \mathbf{B}, \mathbf{S}; \mathbf{X}, \mathbf{Y})$. Again, this model seeks a joint factorization of the data matrix $\mathbf{X}$ and target matrix $\mathbf{Y}$; here both the reconstruction error and supervision error are penalized by the information divergence error function. The multiplicative updates are derived as follows. The gradients of $F_4(\mathbf{A}, \mathbf{B}, \mathbf{S}; \mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$ are, respectively

$$\begin{aligned}
\nabla_{\mathbf{A}} F_4 &= \mathbf{W}\mathbf{S}^\top - \left[\frac{\mathbf{W} \odot \mathbf{X}}{\mathbf{W} \odot \mathbf{AS}} \odot \mathbf{W} \right] \mathbf{S}^\top, \\
\nabla_{\mathbf{B}} F_4 &= \mathbf{L}\mathbf{S}^\top - \left[\frac{\mathbf{L} \odot \mathbf{Y}}{\mathbf{L} \odot \mathbf{BS}} \odot \mathbf{L} \right] \mathbf{S}^\top, \text{ and} \\
\nabla_{\mathbf{S}} F_4 &= -\mathbf{A}^\top \left[\frac{(\mathbf{W} \odot \mathbf{X})}{(\mathbf{W} \odot \mathbf{AS})} \odot \mathbf{W} \right] + \mathbf{A}^\top \mathbf{W} - \lambda \mathbf{B}^\top \left[\frac{(\mathbf{L} \odot \mathbf{Y})}{(\mathbf{L} \odot \mathbf{BS})} \odot \mathbf{L} \right] + \lambda \mathbf{B}^\top \mathbf{L}.
\end{aligned}$$

The multiplicative updates of Algorithm 5 are given by

$$\begin{aligned}
\mathbf{A} &\leftarrow \mathbf{A} - \Gamma \odot \nabla_{\mathbf{A}} F_4 \quad \text{for} \quad \Gamma = \frac{\mathbf{A}}{\mathbf{W}\mathbf{S}^\top}, \\
\mathbf{B} &\leftarrow \mathbf{B} - \Gamma \odot \nabla_{\mathbf{B}} F_4 \quad \text{for} \quad \Gamma = \frac{\mathbf{B}}{\mathbf{L}\mathbf{S}^\top}, \text{ and} \\
\mathbf{S} &\leftarrow \mathbf{S} - \Gamma \odot \nabla_{\mathbf{S}} F_4 \quad \text{for} \quad \Gamma = \frac{\mathbf{S}}{\mathbf{A}^\top\mathbf{W} + \lambda\mathbf{B}^\top\mathbf{L}}.
\end{aligned}$$

```
Algorithm 5: $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF multiplicative updates
Require: $\mathbf{X}, \mathbf{W} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$, $\mathbf{Y}, \mathbf{L} \in \mathbb{R}_{\geq 0}^{k \times n_2}$, $r$, $\lambda$, $N$
  Initialize $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{k \times r}$
  for $i = 1, \ldots, N$ do
    $\mathbf{A} \leftarrow \frac{\mathbf{A}}{\mathbf{W}\mathbf{S}^\top} \odot \left[\frac{(\mathbf{W} \odot \mathbf{X})}{(\mathbf{W} \odot \mathbf{A}\mathbf{S})} \odot \mathbf{W}\right] \mathbf{S}^\top$;
    $\mathbf{B} \leftarrow \frac{\mathbf{B}}{\mathbf{L}\mathbf{S}^\top} \odot \left[\frac{(\mathbf{L} \odot \mathbf{Y})}{(\mathbf{L} \odot \mathbf{B}\mathbf{S})} \odot \mathbf{L}\right] \mathbf{S}^\top$;
    $\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{A}^\top \left[\frac{(\mathbf{W} \odot \mathbf{X})}{(\mathbf{W} \odot \mathbf{A}\mathbf{S})} \odot \mathbf{W}\right] + \lambda \mathbf{B}^\top \left[\frac{(\mathbf{L} \odot \mathbf{Y})}{(\mathbf{L} \odot \mathbf{B}\mathbf{S})} \odot \mathbf{L}\right]}{\mathbf{A}^\top\mathbf{W} + \lambda\mathbf{B}^\top\mathbf{L}}$;
  end for
```

As previously stated in Section 3.2.2, an instance of each family of models, $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF, $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF, $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF, and $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF, correspond to the MLE in the case that the data $\mathbf{X}$ and supervision $\mathbf{Y}$ are sampled from specific distributions with mean given by a latent lower-dimensional factorization model. One might expect that each model is most appropriately applied when the associated model of uncertainty is a reasonable assumption (i.e., one has *a priori* information indicating so). For example, we expect that the Poisson uncertainty assumption in the data reconstruction error associated to $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF or $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF is likely most appropriate when the data features are word counts.

We note that the iteration complexity of each of these methods scales with complexity of multiplication of matrices of size $n_1 \times \max\{k, r\}$ and $\max\{k, r\} \times n_2$. In our implementation of each of these methods, we ensure that there is no division by zero by adding a small positive value to all entries of divisors. Implementations of these methods are available in the Python package `SSNMF` [68]. Finally, we note that the behavior of these models and methods are highly dependent on the hyperparameters r, λ, and N. One can select the parameters according to *a priori* information or use a heuristic selection technique; we use both and indicate selected parameters and method of selection.

3.2.4 Classification Framework

In this section, we describe a framework for using any of the SSNMF models for classification tasks. Given training data $\mathbf{X}_{\text{train}}$ (with any missing data indicated by matrix $\mathbf{W}_{\text{train}}$) and labels $\mathbf{Y}_{\text{train}}$, and testing data $\mathbf{X}_{\text{test}}$ (with unknown data indicated by matrix $\mathbf{W}_{\text{test}}$), we first train our $(R(\cdot\|\cdot), S(\cdot\|\cdot))$-SSNMF model to obtain learned dictionaries $\mathbf{A}_{\text{train}}$ and $\mathbf{B}_{\text{train}}$. We then use these learned matrices to obtain the representation of test data in the subspace spanned by $\mathbf{A}_{\text{train}}$, $\mathbf{S}_{\text{test}}$, and the predicted labels for the test data $\mathbf{Y}_{\text{test}}$. This process is:

1. Compute $\mathbf{A}_{\text{train}}, \mathbf{B}_{\text{train}}, \mathbf{S}_{\text{train}}$ as $\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} R(\mathbf{W}_{\text{train}} \odot \mathbf{X}_{\text{train}}, \mathbf{W}_{\text{train}} \odot \mathbf{AS}) + \lambda S(\mathbf{Y}_{\text{train}}, \mathbf{BS})$.
2. Solve $\mathbf{S}_{\text{test}} = \underset{\mathbf{S}\geq 0}{\operatorname{argmin}} R(\mathbf{W}_{\text{test}} \odot \mathbf{X}_{\text{test}}, \mathbf{W}_{\text{test}} \odot \mathbf{A}_{\text{train}}\mathbf{S})$.
3. Compute predicted labels as $\hat{\mathbf{Y}}_{\text{test}} = \text{label}(\mathbf{B}_{\text{train}}\mathbf{S}_{\text{test}})$, where $\text{label}(\cdot)$ assigns the largest entry of each column to 1 and all other entries to 0 (or more general functions for multi-class classification).

In step 1, we compute $\mathbf{A}_{\text{train}}, \mathbf{B}_{\text{train}}$, and $\mathbf{S}_{\text{train}}$ using implementations of the multiplicative updates methods described above. In step 2, we use either a nonnegative least-squares method (if $R = \|\cdot\|_F$) or one-sided multiplicative updates only updating $\mathbf{S}_{\text{test}}$ (if $R = D(\cdot\|\cdot)$). We note that this framework is significantly different than the classification framework proposed in [104]; in particular, we use the classifier $\mathbf{B}$ learned by SSNMF, rather than independent SVM trained on the SSNMF-learned lower-dimensional representation.

3.3 Numerical Experiments

The 20 Newsgroups dataset [141] is a collection of approximately 20,000 newsgroup documents commonly used as an experimental benchmark for document classification and clustering; see e.g., [104]. The dataset consists of six groups partitioned roughly according to subjects, with a total of 20 subgroups.

Table 3.2: 20 Newsgroups and subgroups.

Groups	Subgroups
Computers	graphics, mac.hardware, windows.x
Sciences	crypt(ography), electronics, space
Politics	guns, mideast
Religion	atheism, christian(ity)
Recreation	autos, baseball, hockey

We consider a subset of the dataset, summarized in Table 3.2. We treat the groups as classes and assign them labels, and we treat the subgroups as (un-labeled) latent topics in the data. [6]

3.3.1 Document Representation

We represent each document by an m-dimensional vector, where m is the number of terms in the dictionary. Terms could consist of n-grams (e.g. unigrams, one-word sequences, or bigrams, two-word sequences).

Let TF_{ij} denotes the frequency of term t_i in document d_j. For a document d_j, the set of weights determined by TF_{ij} may be viewed as a quantitative digest of that document, known in the literature as the *bag-of-words* model. The exact ordering of the terms in a document is ignored; only the number of occurrences of each term is measured

Let DF_i represent the number of documents containing term t_i. Given n documents, we construct a term-document matrix $X \in \mathbb{R}^{m \times n}$ consisting of term-frequency inverse-document-

[6] Our results present this data in its raw form; in particular, we do not capitalize words to reflect common usage. Results are in no way meant to be a political statement.

frequency (TF-IDF) weights as

$$X_{ij} = TF_{ij} \log\left(\frac{n}{DF_i}\right),$$

which represent the significance of term t_i in document d_j [114].

3.3.2 Preprocessing and Choice of Parameters

We remove headers, footers, and quotes from all documents, subsample the dataset to obtain a balanced dataset across classes (1796 document per class), and split the dataset into train (60%), validation (20%), and test (20%) sets. We consider only unigrams and compute the TF-IDF representation for documents using TFIDFVectorizer [131]. The Natural Language Toolkit (NLTK) English stopword list [14], and words appearing in less than 5 documents or more than 70% of the documents were removed. We use the tokenizer `[a-zA-Z]+` and limit the vocabulary size to 5000. We compare to the linear Support Vector Machine (SVM) classifier and Multinomial Naive Bayes (NB) (see e.g., [113]) using the Scikit-learn implementation with default parameters [131], where the groups in Table 3.2 are treated as classes. We consider all SSNMF models with the training process described in Section 3.2.4 with the maximum number of iterations (number of multiplicative updates) $N = 50$. Our stopping criterion is the earlier of N iterations or relative error

$$\frac{F(\mathbf{A}^{(N-1)}, \mathbf{B}^{(N-1)}, \mathbf{S}^{(N-1)}; \mathbf{X}, \mathbf{Y}) - F(\mathbf{A}^{(N)}, \mathbf{B}^{(N)}, \mathbf{S}^{(N)}; \mathbf{X}, \mathbf{Y})}{F(\mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{S}^{(0)}; \mathbf{X}, \mathbf{Y})} \tag{3.10}$$

below tolerance, tol where F denotes the objective function.

We also apply SVM as a classifier to the low-dimensional representation obtained from NMF as follows. We consider the default implementation [131] of $\|\cdot\|_F$-NMF with multiplicative updates, random initialization, and maximum number of iterations $N = 400$. We apply NMF on the training data to obtain a vocabulary dictionary matrix $\mathbf{A}_{\text{train}}$ and a document representation $\mathbf{S}_{\text{train}}$. Next, we train an SVM classifier using $\mathbf{S}_{\text{train}}$ and the labels of the train set. We test our model by (i) computing the document representation of the test data $\mathbf{S}_{\text{test}}$ from the learned dictionary $\mathbf{A}_{\text{train}}$ (i.e., step 2 of Section 3.2.4), and then (ii) applying the trained SVM classifier on $\mathbf{S}_{\text{test}}$ to obtain the test predicted labels.

For both NMF models and all four SSNMF models, we consider rank (the number of topics[7]) equal to 13. We select the hyperparameters tol and λ for the models by searching over different values and selecting those with the highest average classification accuracy on the validation set; see Appendix A.2.

3.3.3 20 Newsgroups Dataset Experiments

We report in Table 3.3 the mean and standard deviation of the test classification accuracy for each of the models over 11 trials. We define the test classification accuracy as $\sum_{i=1}^{n} \delta(\mathbf{Y}_i, \hat{\mathbf{Y}}_i)/n$, where $\delta(u, v) = 1$ for $u = v$, and 0 otherwise, and where $\mathbf{Y}_i$ and $\hat{\mathbf{Y}}_i$ are true and predicted labels, respectively. We observe that $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF produces the highest average classification accuracy, and is comparable to Multinomial NB.

Table 3.3: Mean (and std. dev.) of test classification accuracy for each of the models on the subset of the 20 Newsgroups dataset described in Table 3.2.

Model	Class. accuracy % (sd)
$(\|\cdot\|_F, \|\cdot\|_F)$	79.37 (0.47)
$(\|\cdot\|_F, D(\cdot\|\cdot))$	79.51 (0.38)
$(D(\cdot\|\cdot), \|\cdot\|_F)$	**81.88** (0.44)
$(D(\cdot\|\cdot), D(\cdot\|\cdot))$	81.50 (0.47)
$\|\cdot\|_F$-NMF + SVM	70.99 (2.71)
$D(\cdot\|\cdot)$-NMF + SVM	74.75 (2.50)
SVM	80.70 (0.27)
Multinomial NB	**82.28**

In Table 3.3, we separate models that simultaneously perform dimensionality-reduction and classification from those which only perform classification. Note that the SSNMF models, which provide both dimensionality-reduction and classification in that lower-dimensional

[7] A larger choice of rank could be made to learn hidden topics within subgroups.

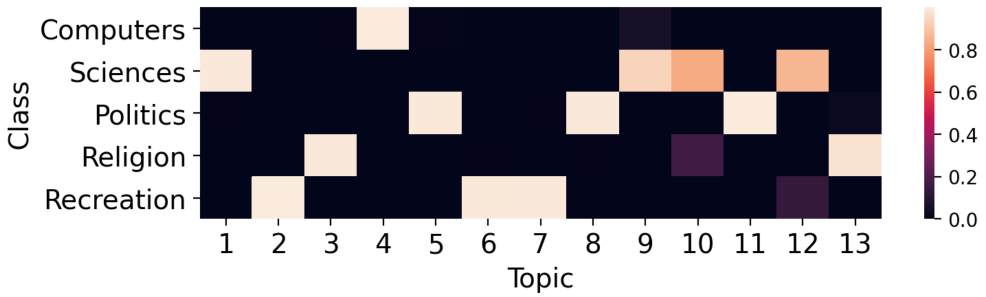

Figure 3.2: The normalized $\mathbf{B}_{\text{train}}$ matrix for the $(D(\cdot\|\cdot), \|\cdot\|_F)$ SSNMF decomposition corresponding to the median test classification accuracy equal to 81.78.

space, do not suffer great accuracy loss which suggests that the simultaneously learned low-dimensional representation serves the classification task well. The SSNMF framework provides an intermediate layer that allows for additional interpretability by representing the data points in the low-dimensional topics space, where we learn about the shared and discriminative topics between classes. This serves the purpose of topic modeling (dimensionality reduction and clustering) and classification. Further, we observe that $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF performs significantly better than $D(\cdot\|\cdot)$-NMF + SVM in terms of accuracy emphasizing the importance of learning simultaneously a linear classifier and a low-dimensional representation. In Appendices A.2 and A.3, we present NMF and SSNMF model results, and compare keywords, classifier matrices, and clustering performance.

Table 3.4: Top keywords representing each topic of the $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF model referred to in Figure 3.2.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
would	game	god	x	would	game	players	people	would	one	israel	like	god
space	team	would	thanks	armenian	one	team	israel	chip	us	guns	anyone	people
government	car	one	anyone	one	like	car	gun	key	get	people	available	church
use	games	jesus	graphics	people	car	last	right	algorithm	could	gun	key	one
key	engine	think	know	fbi	baseball	year	government	use	like	well	probably	christians
chip	year	bible	use	armenians	think	game	us	using	earth	weapons	right	jesus
get	like	believe	mac	israeli	get	hockey	say	bit	space	know	phone	would
clipper	know	christian	please	killed	season	would	jews	like	know	like	another	religious
one	espn	christ	would	fire	last	go	arab	system	see	government	also	christian
could	get	say	get	jews	would	time	one	data	used	would	big	different

Here, we consider the “typical" decomposition for the $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF by selecting the decomposition corresponding to the median test classification accuracy. We display in Figure 3.2 the column-sum normalized $\mathbf{B}_{\text{train}}$ matrix of the decomposition, where each column illustrates the distribution of topic association to classes. We display in Table 3.4 the top 10 keywords (i.e. those that have the highest weight in topic column of $\mathbf{A}_{\text{train}}$) for each topic of the $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF of Figure 3.2.

We (qualitatively) observe from Table 3.4 that topic 5 (“armenian", “fbi"), topic 8 (“arab"), and topic 11 (“weapons") share common keywords (e.g. “israel", “government", “gun") and are associated with class Politics; see Figure 3.2. We also observe that topic 1 (“space", “government") and topic 9 (“chip", “key", “algorithm") are associated to class Sciences. Topic 2 is related to autos (“car", “engine"), topic 6 captures the specific subject of baseball, and topic 7 of hockey. Indeed, all three topics are associated to class Recreation, and topic 12 (“available",“key",“phone") is shared between Sciences and Recreation. Topics 3 and 13 capture topics related to religion and beliefs (“god", “believe", “religious") and are associated to class Religion. Topic 10 (“earth", “space") is shared between Religion and Sciences. Topic 4 captures computer subjects (“x" for Windows 10, “graphics", and “mac"). Indeed, topic 4 is the only topic associated to class Computers in Figure 3.2.

While the learned topics in Table 3.4 are not in one-to-one correspondence with the subgroups in Table 3.2, these topics appear relatively coherent. We see in Table 3.3 that these learned topics serve the classification task well; that is, the data representation in this significantly lower-dimensional space is able to achieve nearly the same accuracy as the higher-dimensional multinomial NB model.

3.4 Conclusion

In this work, we propose SSNMF models which utilize information divergence on the data reconstruction term. This is motivated by count data which is often best described as following a Poisson distribution, which leads to the information divergence in the MLE model [37,74,121]. We demonstrate that these models and that of [104] are MLE in the case of specific distributions of uncertainty assumed on the data and labels. Further, we provide multiplicative update training methods for each model, and demonstrate the ability of these models to perform document classification.

In recent work [69], we further illustrate the promise of these models and training methods on single-label and multi-label document classification datasets (e.g., Reuters, BBC News). In future work, we plan to take a Bayesian approach to SSNMF by assuming data-appropriate priors and performing maximum *a posteriori* estimation. Further, we will form a general framework of MLE models for exponential family distributions of uncertainty, and study the class of models where multiplicative update methods are feasible.

Chapter 4

Detecting Short-Lasting Topics Using Nonnegative Tensor Decomposition

4.1 Introduction

Topic modeling is an unsupervised machine learning technique used to reveal latent themes from large text datasets. *Dynamic topic modeling* investigates how topics evolve in a sequentially organized corpus of documents, where the data is typically divided by time slices [16, 80, 81, 145]. Temporal text data, such as news articles or Twitter feeds, often consists of a mixture of long-lasting trends and popular but short-lasting topics of interest. A truly successful topic modeling strategy should be able to detect both types of topics and clearly locate them in time. However, we find that Latent Dirichlet Allocation (LDA) and Nonnegative Matrix Factorization (NMF), two popular classic methods in topic modeling, do not detect such short-lasting topics on real-world datasets of interest when the temporal data is aggregated along the time dimension. On the contrary, we find that nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD) [27, 71] can successfully detect transient topics on semi-synthetic and real-world text datasets. In LDA, one models a topic by a probability distribution on the set of words, which are evolved according to a Bayesian scheme by feeding in subsequent time slices [16, 76]. On the other hand, there are two basic methods of using NMF for dynamic topic modeling. First, one can factorize each time slice independently using NMF [3, 101, 103, 124]. Second, one can first concatenate all the time slices along the documents dimension and decompose the resulting matrix using NMF with a fixed dictionary to obtain common topics; the evolution of such topics is then found by computing their contribution in each time slice [42].

We note that it is natural to encode a sequence of documents as a 3-dimensional tensor, a common algebraic representation for high-dimensional arrays, where the three modes correspond to words, documents, and time, respectively. The crucial step of (dynamic) topic modeling is to decompose high-dimensional data (tensors) into interpretable representations with attention to the temporal information. In addition, one may also be interested in finding such decompositions with some additional structure, such as nonnegativity, which allows for interpretability of topics [1] as opposed to traditional matrix factorization approaches like principal component analysis (PCA) where factors often cancel due to subtractive combinations [103].

The proposed method based on nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD) [27,71] processes the entire 3-dimensional tensor at once by finding three *nonnegative* factor matrices of shape ($\texttt{words} \times \texttt{topics}$), ($\texttt{documents} \times \texttt{topics}$), and ($\texttt{time} \times \texttt{topics}$) such that their outer product approximates the original 3-dimensional tensor. Note that NCPD for 2-dimensional tensors is equivalent to the nonnegative matrix factorization (NMF) from Chapter 3, which is well-known to be able to extract spatially localized features when applied to image data [103]. Roughly speaking, being a 3-dimensional analogue of NMF, NCPD is able to extract spatio-temporally localized features, where 'spatial localization' in our context means words that form topics, which are also transient by 'temporal localization'. We also note that one of the advantages of NCPD and NMF over existing LDA methods is that there are fewer parameter choices involved in the modeling process. In our experiments, we show that NCPD successfully detects short-lasting topics that other popular methods such as LDA and NMF fail to fully detect. We demonstrate the ability of NCPD to discover short and long-lasting temporal topics in semi-synthetic and real-world data including news headlines and COVID-19 related tweets.

4.1.1 Background and Related Work

In this section, we describe related work in three main areas: dynamic topic modeling, applications of tensor decompositions, and studies on COVID-19 related tweets.

Dynamic Topic Modeling

Several works have examined topics and their evolution through time using probabilistic models [16, 163], nonnegative matrix factorizations [9, 42, 63], and deep learning models [126]. In [16], Blei and Lafferty propose a generative model which is an extension to Latent Dirichlet Allocation (LDA) to handle a sequentially organized corpus of documents. The applicability of the dynamic topic model is demonstrated by analyzing over 100 years of articles from the journal *Science* aiming to show that this method can be used to analyze the trends of word usage inside topics. In [163], the authors propose a continuous time dynamic topic model (cDTM) which uses Brownian motion to model latent topics through a sequential collection of documents, where a “topic” is a pattern of word use that is expected to evolve over the course of the collection.

Applications of Tensor Decompositions

Tensor decompositions have many applications in machine learning [95, 134] including temporal analysis such as discovering patterns [167], discovering time-evolving topics [4, 5, 95], predicting evolution [47], modeling the behaviors of drug-target-disease interactions [31], detecting time-evolving phenotypic topics [174], spotting anomalies [125], and identifying fake news [78]. Tensor methods have also been successfully applied to independent component analysis [8] and probabilistic topic models [2]. We refer the reader to [95, 134] for surveys that provide an overview of higher-order tensor decompositions and their applications.

The most related work is [4], where the authors use a nonnegative PARAFAC tensor factorization to analyze Enron email data. They demonstrate that the approach provides more interpretability than nonnegative methods, but do not analyze structural or temporal differences between matrix or tensor variants. In particular, they group emails by month, and comment that it is interesting future work to study the effect of varying this granularity. Here, we uncover the important phenomenon that topics that are short lasting are revealed more readily using NCPD than by using the matrix variants, suggesting yet another advantage of utilizing higher mode structures.

COVID-19 Related Tweets

Analyzing social media using various dynamic topic models has become popular for studying and tracking various public health events around the world [32, 128, 129]. In this work, we consider data from the social media platform, Twitter. During the COVID-19 pandemic, Twitter has experienced increased usage including discussion and dissemination of information relating to the pandemic [30, 158]. A number of related works consider Twitter and other social-media data related to the COVID-19 pandemic via various statistical and learning approaches, with specific aims to understand the effects and prevalence of bots and misinformation [52, 170], polarization [18, 62], sentiment and emotional state [94, 172], gender differences [159], racism and xenophobia [175], politics [148], and other aspects [123].

4.1.2 Contribution

Investigating how latent themes emerge, evolve, and fade in temporally dynamic text datasets may provide valuable insights in understanding both large-scale trends and impactful shorter-lasting events. Nonnegative factorizations are used ubiquitously for topic identification and interpretability. However, there is less work that makes use of NCPD for this purpose, making it ever more important to study the differences in output when using a matrix versus a tensor factorization method with temporal data. While some major topics may persist for an extended period of time, detecting *short-lasting topics*, that correspond to shorter-lasting, but impactful events or discussions, is equally important. In all experiments, we find that NCPD successfully detects both short-lasting and longer-lasting topics, whereas LDA- and NMF-based methods primarily detect only long-lasting topics. The main contributions of this work are as follows:

- **Detect short-lasting topics.** We demonstrate nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD) as a dynamic topic modeling technique able to detect and accurately represent both long- and short-lasting topics from temporal text data.
- **Compare with traditional matrix methods.** We show that NCPD performs significantly better than standard matrix-based topic modeling methods such as LDA and NMF in de-

tecting topics with short durations on semi-synthetic 20 Newsgroups dataset [141] designed as a benchmark dynamic text dataset, and two real-world datasets: news headlines and COVID-19 related tweets.

- **COVID-19 related tweets.** An interesting auxiliary result of this work is the analysis of Twitter text data related to the COVID-19 pandemic [30]. From the text data, the methods discover topics trending in news sources, including political events, personal beliefs about COVID-19, and calls to action, and successfully attribute the topics to the days they were trending[8].

Our code for reproducing experiments is publicly available[9].

4.1.3 Organization

In Section 4.2, we describe the dynamic topic modeling methods considered in this work, namely NMF, LDA, and NCPD. We present numerical experiments comparing the performance of these methods on (i) semi-synthetic 20 Newsgroups dataset in Section 4.3.2; (ii) COVID-19 related tweets in Section 4.3.3; and (iii) ABC news headlines dataset in Section 4.3.4. Lastly, we end with a conclusion and discussion of future work in Section 4.4.

4.2 Methods for Dynamic Topic Modeling

In this section, we introduce notation and discuss the dynamic topic modeling methods NMF, LDA, and NCPD.

4.2.1 Notation

We denote third-order tensors with uppercase calligraphic letters $\mathscr{X}$. *Tensors* are common algebraic representations for multidimensional arrays. The *order* of a tensor is the number of

[8] The topics learned from data and news references may contain factually inaccurate information. While we include some news references to validate results, many additional sources exist. We make no claims that these sources are factually accurate.

[9] https://github.com/lara-kassab/dynamic-tensor-topic-modeling

dimensions, which is also referred to as *ways* or *modes* [95]. For a matrix $\mathbf{X}$, the vector x_k denotes its k^{th} column. We let $\|\cdot\|_F$ and $\|\cdot\|_1$ denote the entrywise Frobenius norm, and the entrywise L_1 norm, respectively. The set of nonnegative real numbers $[0,\infty)$ is denoted $\mathbb{R}_{\geq 0}$. See [95] for an excellent survey of related definitions and algorithms for tensor decomposition.

4.2.2 Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) is a popular tool for extracting hidden themes from text data [20, 98]. For a data matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$, one learns a low-rank dictionary $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$ and representation matrix $\mathbf{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$ that minimize $\|\mathbf{X} - \mathbf{AS}\|_F^2$, where $r > 0$ is typically chosen such that $r < \min\{n_1, n_2\}$. Suppose n_1 denotes the number of features (in our case unigrams and bigrams; see e.g. 3.3.1) and n_2 the number of documents, then the dictionary matrix $\mathbf{A}$ represents *topics* in terms of the original features. Each column of the representation matrix $\mathbf{S}$ represents a data point as a linear combination of the dictionary elements with nonnegative coefficients.

To extract temporal information from a basic NMF decomposition, we perform the following [42]. We concatenate $\mathbf{X}_i \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$ for each time slice $1 \leq i \leq n_3$ along the columns to obtain $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times (n_2 \cdot n_3)}$ where $\mathbf{X} = \left[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{n_3}\right]$. We use NMF to learn a dictionary matrix $\mathbf{A}$ from all the time slices,

$$\left[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{n_3}\right] \approx \mathbf{A}\left[\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{n_3}\right],$$

where $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$, and $\mathbf{S}_i \in \mathbb{R}_{\geq 0}^{r \times n_2}$ is the representation matrix for each time slice $i = 1, \cdots, n_3$. We analyze topic dynamics through changes in topic prevalence over time in the representation matrices. We compute the mean topic representation $\bar{s}_i \in \mathbb{R}_{\geq 0}^{r \times 1}$ for each time slice i by taking the average over the columns of the matrix $\mathbf{S}_i$. We present $\bar{s}_i$ as the columns of the heatmaps (e.g. Figure 4.3).

4.2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is another popular tool for extracting hidden topics from text data. LDA is a generative probabilistic bag-of-words model of a corpus, where documents are represented as random mixtures over latent topics, and where each topic is characterized by a distribution over words [17]. We summarize the generative process of LDA in Algorithm 4.1 (see e.g. [17, 73]), and we detail all the quantities of the model in Table 4.1.

Algorithm 6: Generative model for latent Dirichlet allocation

```
for all topics $k \in [1, K]$ do
    sample mixture components $\phi_k \sim Dir(\beta)$
end for
for all documents $m \in [1, M]$ do
    sample mixture proportion $\theta_m \sim Dir(\alpha)$
    sample document length $N_m \sim Poiss(\zeta)$
    for all words $n \in [1, N_m]$ in document $m$ do
        sample topic index $z_{m,n} \sim Mult(\theta_m)$
        sample term for word $w_{m,n} \sim Mult(\phi_{z_{m,n}})$
    end for
end for
```

Table 4.1: Quantities in the model of latent Dirichlet allocation [73].

M	number of documents in the corpus
K	number of topics/mixture components
V	number of terms t in vocabulary
α	hyperparameter on the mixing proportions (K-vector or scalar if symmetric)
β	hyperparameter on the mixture components (V-vector or scalar if symmetric)
θ_m	parameter notation for $p(z\|d = m)$, the topic mixture proportion for document m. One proportion for each document, $\Theta = \{\theta_m\}_{m=1}^{M}$ ($M \times K$ matrix)
ϕ_k	parameter notation for $p(t\|z = k)$, the mixture component of topic k. One component for each topic, $\Phi = \{\phi_k\}_{k=1}^{K}$ ($K \times V$ matrix)
N_m	document length (document-specific), here modelled with a Poisson distribution with constant parameter ζ
$z_{m,n}$	mixture indicator that chooses the topic for the nth word in document m
$w_{m,n}$	term indicator for the nth word in document m

The intuition behind LDA is that documents exhibit multiple topics. The model assumes that the document at first is empty, then choose a topic from the topic mixture and then a word from the word mixture of that topic, repeating this process until the document is shaped [39]. This process is repeated for every document in the corpus and assumes that the order of the words in the documents does not necessarily matter.

In general, there are two approaches that are commonly used to approximate posterior distributions, Markov Chain Monte Carlo (MCMC) methods and variational inference. In our experiments, we consider symmetric prior distributions and an LDA model that uses online variational inference [76]. We analyze topic dynamics through changes in topic prevalence over time in the Θ matrix. We compute the mean topic representation for each time slice i by taking the average over the row of the matrix Θ_i, where Θ_i denotes the ith time slice of Θ. We present the mean topic representation as the columns of the heatmaps (e.g. Figure 4.4).

4.2.4 Nonnegative CP Tensor Decomposition

Nonnegative CP Tensor Decomposition (NCPD) is a tool for decomposing higher-dimensional data tensors into interpretable lower-dimensional representations. NCPD factorizes a tensor into a sum of nonnegative component rank-one tensors, defined as outer products of nonnegative vectors [27, 71]. More precisely, given a third-order tensor $\mathscr{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2 \times n_3}$ and a fixed integer $r > 0$, the approximate NCPD of $\mathscr{X}$ seeks matrices $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{n_2 \times r}, \mathbf{C} \in \mathbb{R}_{\geq 0}^{n_3 \times r}$, such that $\mathscr{X} \approx \sum_{k=1}^{r} a_k \otimes b_k \otimes c_k$, where the nonnegative vectors a_k, b_k, and c_k are the columns of $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$, respectively. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are referred to as the NCPD *factor matrices.* Such factor matrices are found by solving the following minimization problem

$$\operatorname*{argmin}_{\mathbf{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{n_2 \times r}, \mathbf{C} \in \mathbb{R}_{\geq 0}^{n_3 \times r}} \left\| \mathscr{X} - \sum_{k=1}^{r} a_k \otimes b_k \otimes c_k \right\|_F . \tag{4.1}$$

Note that (4.1) is a non-convex optimization problem, but it is convex for each factor matrix while the other two factors are held fixed. Leveraging this observation, many proposed algorithms for solving (4.1) have the nature of block coordinate descent [13, 166], including the multiplicative update algorithm [150], alternating least squares [27, 71] and its variants [111].

NCPD is considered as a topic modeling technique for tensor data that successfully showcases topic variation across all modes of the tensor (including temporal mode(s)) (see e.g., [1, 4, 134]). Namely, suppose we have a third-order tensor data $\mathscr{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2 \times n_3}$ where $n_1 = \texttt{words}$ denotes the number of words in the vocabulary, $n_2 = \texttt{docs}$ denotes the number of documents, and $n_3 = \texttt{time}$ denotes the number of time slices. Applying NCPD to the third-order tensor data $\mathscr{X}$, we obtain three factor matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ of shapes $(\texttt{words} \times r)$, $(\texttt{docs} \times r)$, and $(\texttt{time} \times r)$, respectively, where $r = \texttt{topics}$ equals the number of topics we seek to find. We will be the most interested in the factor matrices $\mathbf{A}$ and $\mathbf{C}$; the columns of $\mathbf{A}$ give word representation of the topics whereas the corresponding columns of $\mathbf{C}$ give how their importance evolves through the time, i.e. the time representation of the topics. The second factor matrix $\mathbf{B}$ gives the document representation of topics, which is of less importance for our purpose of dynamic topic modeling. In the heatmaps (e.g., Figure 4.5), we present the factor matrix $\mathbf{C}$ and the top keywords for each topic obtained from $\mathbf{A}$.

4.3 Experiments

In this section, we compare the performance of NMF, LDA, and NCPD methods in identifying temporal topics in several datasets. Section 4.3.2 compares the methods on a semi-synthetic dataset derived from the popular 20 Newsgroups dataset [141] to serve as a benchmark. Section 4.3.3 considers a dataset of the top retweeted tweets from each day during several months of the COVID-19 pandemic, which was a particularly dynamic period in terms of the amount of events happening over short periods of time. Finally, we consider a dataset of news headlines from the years 2003 to 2019 that contains long-lasting, short-lasting, and periodic topics in Section 4.3.4. The keyword representation of each of the extracted topics is also provided for interpretability[10]. The number of topics for the semi-synthetic 20 Newsgroups dataset is chosen to match the known number of article subjects. For the real-world COVID-19 Twitter and News Headlines datasets, we choose the number of topics to balance readability and the discovery of relevant events.

4.3.1 Experimental Setup

In all the experiments, documents are converted to term frequency–inverse document frequency (TF-IDF) vector representations (e.g., see Section 3.3.1) using the sklearn TFIDFVectorizer [131]. We compute NMF of the data matrix using sklearn [131] with nonnegative double singular value decomposition initialization [19]. We compute NCPD of the tensor data with multiplicative updates [150] using TensorLy [97] and singular value decomposition initialization. Lastly, for LDA we construct a bag-of-words corpus using the same dictionary as the other methods (obtained from the TF-IDF weights) and compute the model using *Gensim* LDA model [140] with various numbers of passes and training chunks to save memory on larger datasets [76]. Our code is publicly available[11].

4.3.2 Semi-synthetic Dynamic Dataset Results

20 Newsgroups dataset [141] is a collection of documents divided into six groups partitioned into subjects, with a total of 20 subgroups. This dataset is commonly used as an experimental benchmark for document classification and clustering;; see e.g., [55, 75]. We consider a semi-synthetic dataset constructed from the 20 Newsgroups dataset to illustrate the dynamic topic modeling performance of NMF, LDA, and NCPD on a simple and well-understood dataset.

We consider only five categories: “Atheism", “Space", “Baseball", “For Sale", and “Windows X" with a total of 1040 documents. We remove headers, footers, and quotes from all documents

[10] Each learned topic is represented by a positive linear combination of terms. Terms with larger values in a particular topic are more significant for that topic and, thus, the terms with the largest values provide interpretable descriptions of the topics.

[11] https://github.com/lara-kassab/dynamic-tensor-topic-modeling

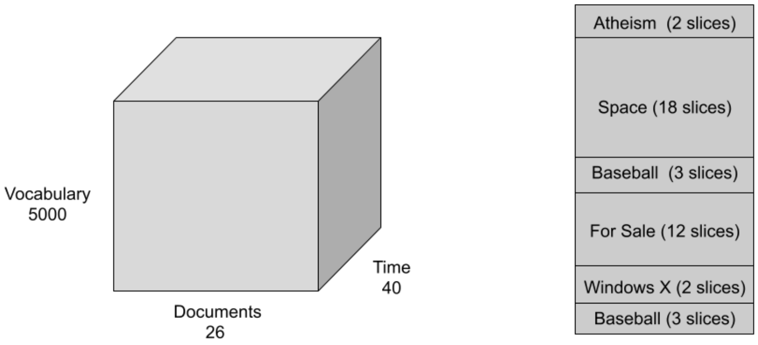

Figure 4.1: Semi-synthetic 20 Newsgroups tensor construction.

and compute the TF-IDF representations of all the documents in the corpus with vocabulary size equal to 5000. The NLTK English stopword list [14], and words appearing in more than 95% of the documents are removed. We organize the dataset into a 5000 × 26 × 40 tensor with dimensions: vocabulary size by number of documents by time. Each time slice consists entirely of articles from the same category, and the categories of the times slices are ordered as: (“Aethism", time slices 1-2), (“Space", time slices 3-20), (“Baseball", time slices 21-23), (“For Sale", time slices 24-35), (“Windows X", time slices 36-37), and (“Baseball", time slices 37-40). We run NMF, LDA, and NCPD as described in Section 4.2 with rank equal to 5 reflecting the number of categories in the dataset. Learned topics and prevalence of each topic over time are indicated for each method.

On this semi-synthetic data, NCPD identifies topics associated with each subject and accurately indicates the temporal occurrence of each subject, while NMF and LDA learn topics that are prevalent during time slices associated with multiple subjects. NCPD learns a single topic for each subject included in the dataset and accurately attributes highest prevalence to the true underlying topic in each time slice. NMF and LDA also learn reasonable topics, including topics corresponding to the longer-lasting “Space" and “For Sale" segments. On this relatively simpler semi-synthetic data, NMF and LDA detect some but not all of the short-lasting topics. For example, NMF's learned topic 1 spikes in prevalence during the short-lasting “Aetheism" and “Baseball" segments, while LDA accurately detects a short-lasting “Windows X" related topic. Both LDA and NMF learn topics that blend multiple document subjects. For example, for both NMF

	NMF	NCPD	LDA
1	would, like, think	space, would, like	would, like, one
2	drive, sale, offer	00, sale, drive	edu, use, window
3	space, shuttle, nasa	games, game, year	space, launch, nasa
4	00, 20, 50	god, believe, religion	new, sale, please
5	mac, hm, msu	window, widget, application	00, 50, 20

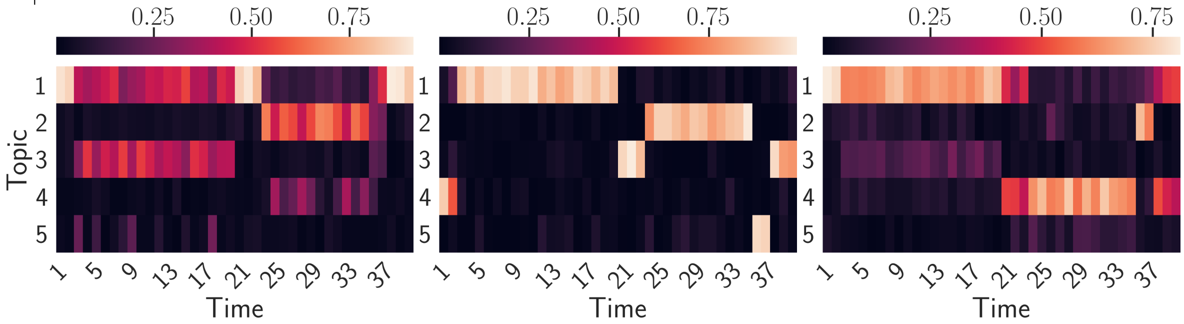

Figure 4.2: The learned topics and prevalence of each extracted topic from the semi-synthetic 20 Newsgroups dataset are shown for each of the three models (NMF, NCPD, LDA). The columns of each heatmap indicate the distribution over the extracted topics for each time slice. The top three keywords corresponding to each topic of the models are provided in the table.

and LDA, the most prevalent topic detected during the “Aetheism" time slices is also present during the “Space" time slices. The difference in the ability to detect short-lasting topics between NCPD versus NMF and LDA is even more drastic for the more complex Twitter and News Headlines datasets (Sections 4.3.3 and 4.3.4).

4.3.3 COVID-19 Related Tweets Dataset Results

The next dataset we consider consists of tweets related to the COVID-19 pandemic [30], collected from February-April 2020. We subsample the dataset to 90,000 tweets by keeping the 1000 most popular tweets per day, so that the number of tweets is the same for each day (“popularity" is measured by the number of retweets)[12]. We limit the vocabulary size to 5000, remove the NLTK English stopword list [14] and non-word sequences such as “https". We additionally remove words that are essentially synonymous with COVID-19 (such as,“coronavirus" or “covid")

[12] We would like to note that online versions of the methods can be used to efficiently process larger datasets; in particular, they can be used to avoid subsampling in the Twitter (Section 4.3.3) and Headlines (Section 4.3.4) dataset.

as all the tweets in the dataset are related to this common topic. We run all the methods with target rank 20 to balance readability and the discovery of relevant events.

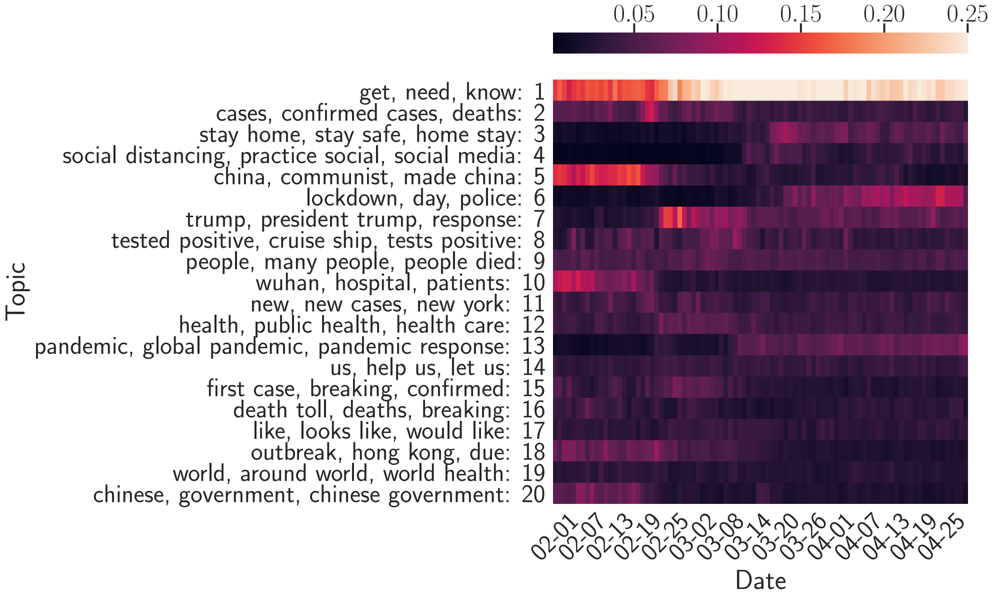

Figure 4.3: The normalized mean topic representation of tweets per day learned via NMF with rank 20.

We present results in the form of heatmaps that summarize both the term representation and temporal prevalence of topics in Figures 4.3 – 4.5. For NMF and LDA, the mean topic representation for each day is given in the columns of the heatmaps, while for NCPD, the factor matrix showcasing the temporal representation of the topics is shown. Each row of the heatmaps corresponds to a learned topic and a three-term summary of each topic is included for interpretability. The top three terms (unigrams and bigrams) are listed for each topic. Each of the methods recovers common large trends in the data. Generally, China-related topics are most prominent in early and mid February. The prevalence of these topics then decreases in mid February. A topic relating to new cases spikes in prevalence in mid February as outbreaks begin to occur around the world. In late February to mid March, a topic relating to U.S. President Trump and his administration's response spikes in prevalence. Separate “social distancing", “stay home", and “lockdown" topics begin in early to mid March. These topics typically persist throughout April. On the other hand, certain topics are only present for a short period of time,

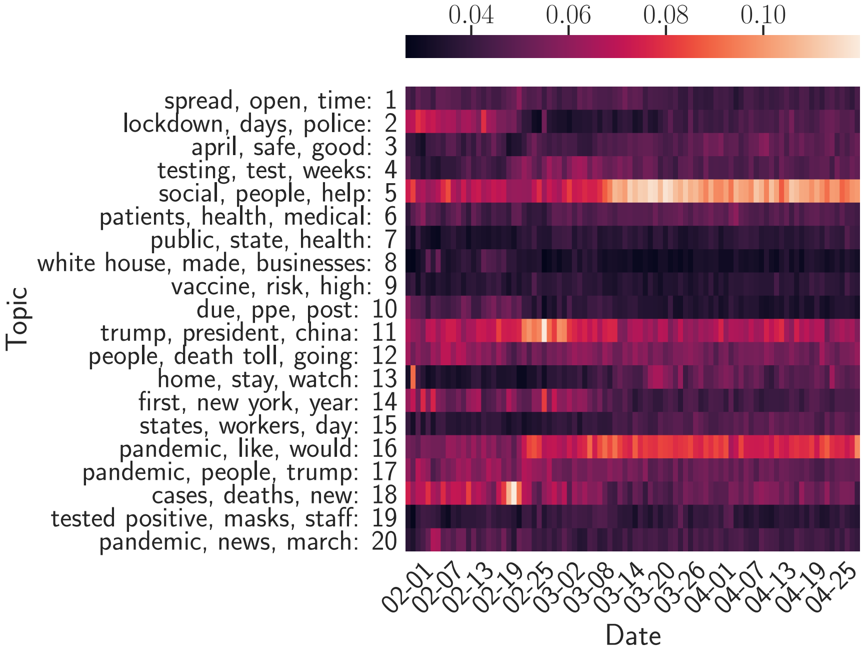

Figure 4.4: The mean topic representation of tweets per day learned via LDA with rank 20.

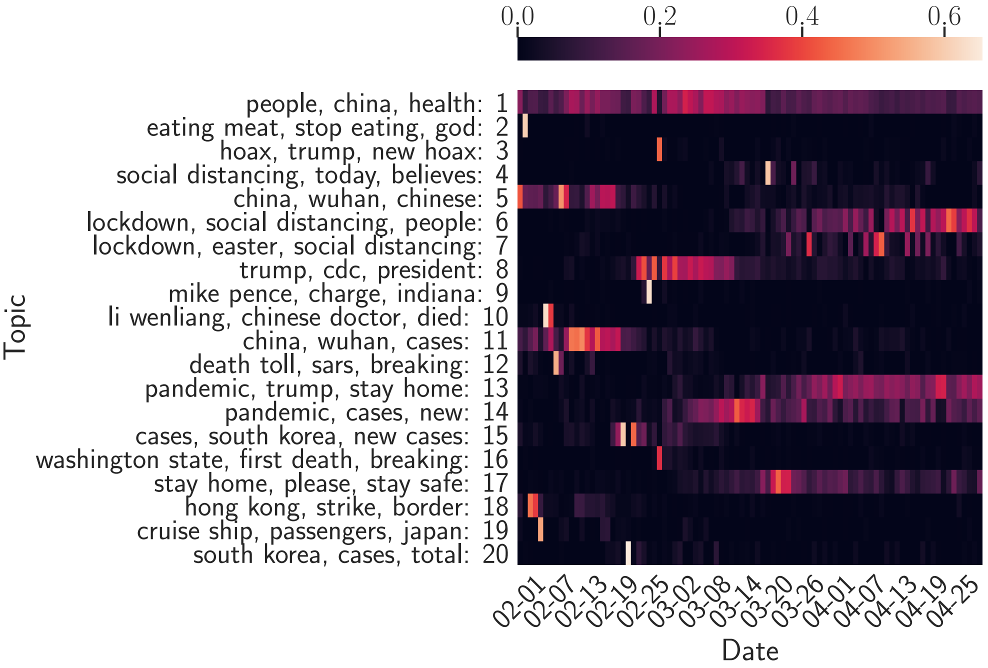

Figure 4.5: The normalized factor matrix of NCPD on the tweets dataset with rank 20.

and such topics were discovered by NCPD only (see Figure 4.5). Several such short-lasting topics are detected by NCPD in February (Figure 4.5). For example, topic 2 relating to the beliefs surrounding eating meat and COVID-19 peaks on February 2 ([165]), and topic 19 related to the passengers of the cruise ship, Diamond Princess, peaks on February 5 ([86]). Further, political topics such as topic 3 relating to President Trump's claims of COVID-19 being the Democrats' new hoax peaks on February 28 ([119]), and topic 9 relating to Vice President Mike Pence's appointment as chair of the White House Coronavirus Task Force peaks on February 26 ([132]). Other topics related to deaths include topic 10 on the death of the Chinese doctor, Dr. Li Wenliang ([7]), which peaks on February 6, and topic 16 related to the first death in Washington State due to COVID-19, which peaks on February 28 (compare with [120]). Lastly, topics related to South Korea include topic 15 and 20, which are most prominent in February 20-22 ([156]). Topic 20 captures "cases/new cases" events related to the outbreak in South Korea more generally, whereas topic 15 captures the outbreak event in Iran.

The fact that the short-lasting topics are discovered at similar times to related news articles additionally demonstrates how quickly events are discussed on Twitter. We also observe that in the first months of the pandemic general Twitter discussion was less homogeneous, leading to greater variety of popular short-lasting topics appearing. Later on, primarily longer-lasting topics are detected by all methods.

4.3.4 News Headlines Dataset Results

A Million News Headlines is a dataset containing news headlines published over a period of 17 years sourced from the Australian news source ABC [99]. The dataset includes noteworthy global events from February-2003 to December-2019 (203 months total) with a focus on Australia. We consider 700 headlines randomly selected per month with a total of 142,100 headlines in the entire dataset. We compute a TF-IDF representation for documents, and limit the vocabulary size to 7000. The NLTK English stopword list [14], "abc", and words appearing in more than 70% of the documents or less than 5 documents were removed. We run NMF, LDA, and NCPD with rank equal to 25 to balance readability and the discovery of relevant events. The dataset contains short-lasting and long-lasting topics along side a temporal structure of *periodic* topics (which we also find much easier to discover with NCPD).

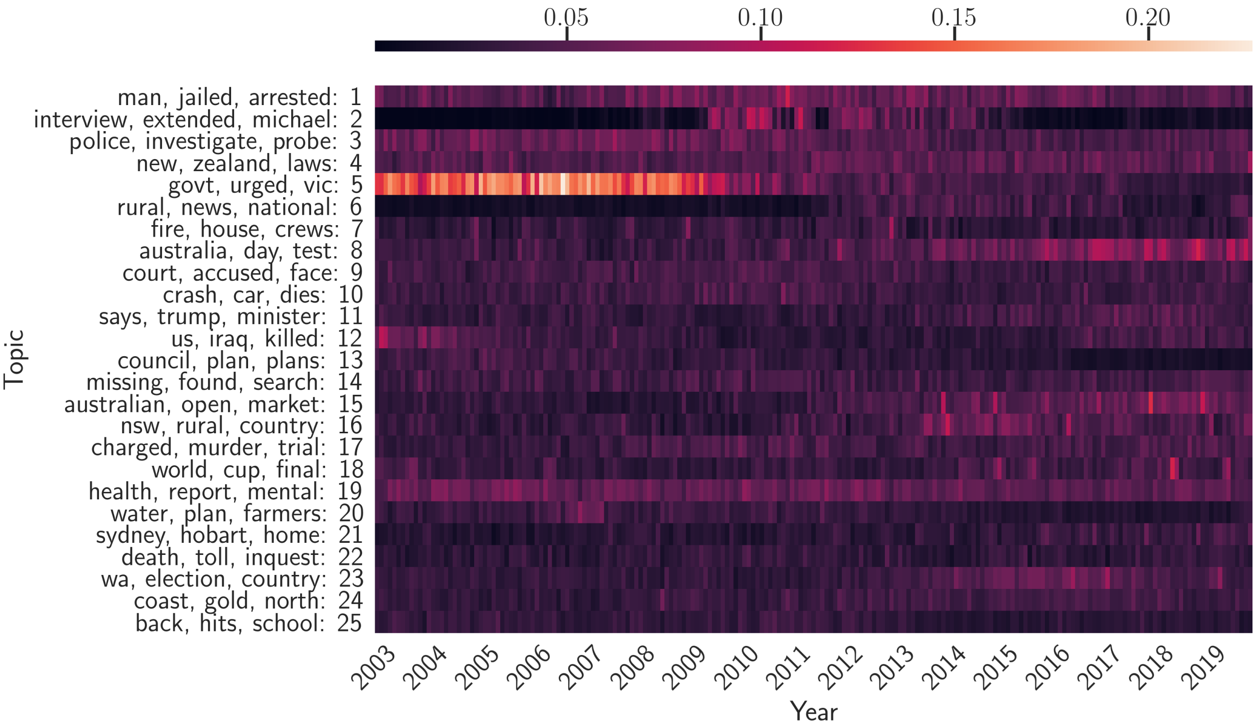

Figure 4.6: The normalized mean topic representation of headlines per month learned via NMF with rank 25.

All of the methods feature several persistent topics, including those trending in earlier and later years only. The majority of such topics are generic and refer to the state entities, society, or police, rather than specific events, e.g., NMF topic 15 (“australian, open, market"), LDA topic 18 (“man, court, murder"), or NCPD, topic 4 (“police, new, us"). There are common topics among them, such as the one related to the government, including NMF, topic 5 (“govt, urged, vic"), LDA, topic 11 (“health, report, govt"), and NCPD, topic 13 (“govt, closer, pm").

Furthermore, we observe that NMF picks up only the topics that persist through time, as we observed in the previous sections. Some of the topics discovered by LDA seem to refer to more time-localized events, such as, LDA, topic 6 (“found, canberra, dead"). However, each topic discovered by LDA includes articles throughout the whole time span.

On the other hand, NCPD for topic modeling (Figure 4.8) discovers a range of short-lasting events, such as the swine flu outbreak peaking in Spring 2019 ([77]) (topic 20, “swine, flu, case"), federal elections in 2009 (topic 24, “election, federal, 2019"), the “Boxing day tsunami" on De-

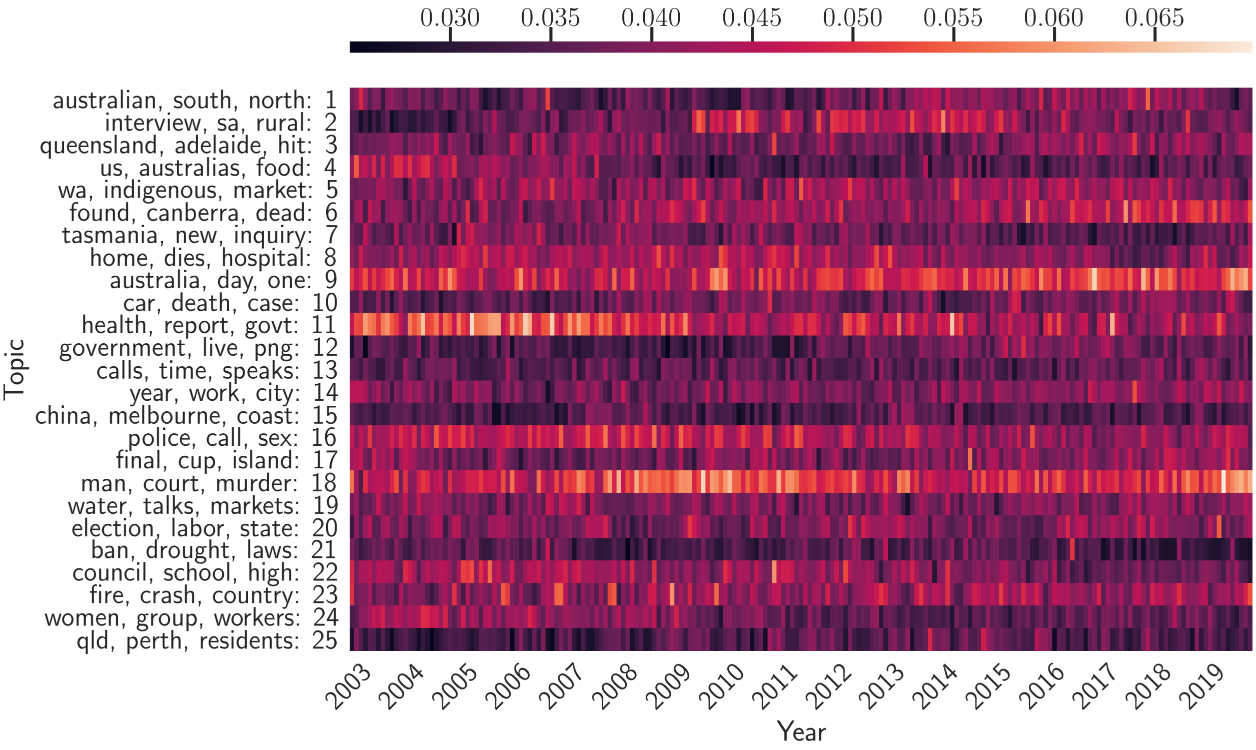

Figure 4.7: The mean topic representation of headlines per month learned via LDA with rank 25.

cember 26th, 2004 ([161]) (topic 8, “tsunami, aid, toll"), and others. Moreover, some learned topics trend periodically throughout the years, being related to a specific time of the year. For example, topic 9 (“budget, federal, may") trends every spring, but especially in 2014, while Boxing day shopping events are present each December (topic 21, “christmas, day, boxing"). We conclude the discussion of the NCPD results with a comparison between the topics 1 and 2, both having “interview" as the most frequent word. Topic 1 (“interview, police, man") refers to more generic police-suspect interviews and is prevalent across several years, whereas topic 2 (“interview, nathan, john") likely refers to popular interview(s) with public figures.

4.4 Conclusion

We demonstrate nonnegative CANDECOMP/PARAFAC decomposition (NCPD) as a powerful dynamic topic modeling technique capable of detecting short-lasting and periodic topics along with long-lasting topics in dynamic text datasets (including news headlines and Twitter feeds). We compare NCPD to other popular dynamic modeling techniques based on NMF and

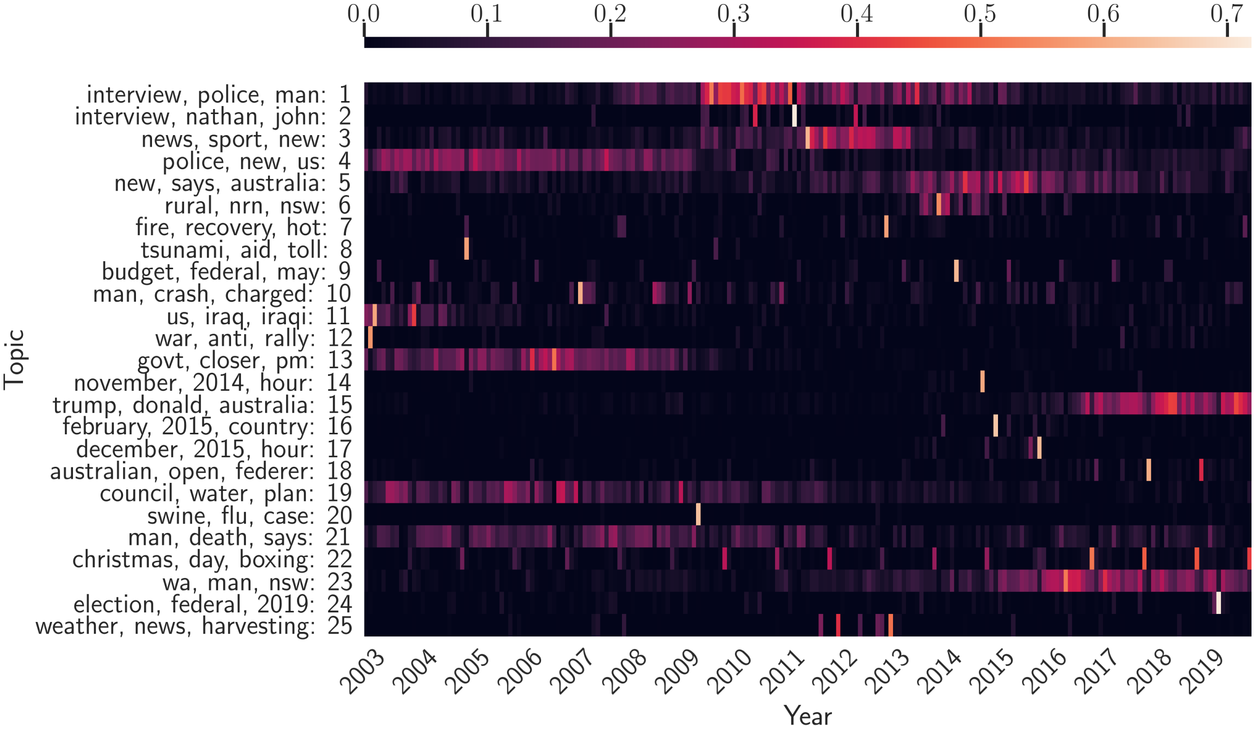

Figure 4.8: The normalized factor matrix of NCPD on the News Headlines dataset with rank 25.

LDA where temporal data is aggregated along the time dimension. We observe that on the simpler semi-synthetic dataset all methods are able to detect short and long-lasting topics, while NCPD is the only method able to detect and accurately represent short-lasting topics in the COVID-19 Twitter dataset and the short-lasting and periodic topics in the headlines dataset. We discuss and compare the temporal topic patterns learned through each of these methods. We validate some learned topics against news sources, and show that NCPD accurately discovers topics trending in news sources, and successfully attributes them to the period of times they were trending.

In recent work [89], we demonstrate an online version of the NCPD algorithm as a method to reduce computation time for large-scale tensors while preserving the temporal patterns of the topics. For large datasets, online NCPD serves as a viable alternative for learning topics and their temporal patterns, retaining the ability to detect short-lasting topics. Further, we propose a quantitative measure of the topic length and numerically demonstrate the variability of the topics discovered by NCPD (as well as its online variant).

Chapter 5

Conclusion

In this dissertation, we propose three techniques that take into account underlying structure in large-scale data to produce better or more interpretable results for machine learning tasks. The first technique is an iterative method for matrix completion designed to handle large-scale datasets and take into account sparsity structure in the missing values to improve recovery. The second is a semi-supervised nonnegative matrix factorization formulation with information divergence as an error function to better model count data and learn a low-dimensional representation that serves a (semi-)supervised machine learning task. The third is a dynamic topic modeling technique that uses nonnegative tensor decomposition to simultaneously process all the modes of the data tensor (e.g. vocabulary, documents, time), resulting in a more time-localized lower-dimensional representation than traditional matrix methods.

In Chapter 2, we describe our work [88] on iterative methods for matrix completion with sparsity-based structure in the missing entries whereby the vector of missing entries is close in the ℓ_0 or ℓ_1 norm sense to the zero vector (or more generally, to a constant vector). We adapt an iterative algorithm for low-rank matrix completion to take into account sparsity-based structure in unobserved entries by adjusting the IRLS-p algorithm studied in [117]. We also present a gradient-projection-based implementation, called *Structured sIRLS* (motivated by sIRLS in [117]). Many research directions remain open in terms of defining matrix completion algorithms for more general structures in the missing entries. This includes the case where the probability that an entry is observed or not may depend on more than just the value of that entry; perhaps on the row and/or column the missing entry belongs to. Other interesting directions include extending such methods to higher-order tensors with certain underlying low-rank and sparsity structures.

In Chapter 3, we describe our work [69] on semi-supervised nonnegative matrix factorization (SSNMF) for learning tasks which use utilize information divergence as an error function.

This is motivated by count data which is often best described as following a Poisson distribution, which leads to the information divergence in the MLE model [37, 74, 121]. We provide motivation for these models as maximum likelihood estimators, derive training methods using multiplicative updates for each new model, and demonstrate the application of these models document classification (e.g., 20 Newsgroups dataset). Interesting future directions include adapting the algorithm for other (semi-)supervised learning tasks such as regression and generalizing such algorithms for higher-order tensors.

In Chapter 4, we describe our work [89] on dynamic topic modeling for temporal text dataset using nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD). We show that NCPD successfully detects short-lasting topics in dynamic text datasets that other popular methods such as latent Dirichlet allocation (LDA) and nonnegative matrix factorization (NMF) fail to fully detect. We demonstrate the ability of NCPD to discover short, long-lasting, and periodic temporal topics in semi-synthetic and real-world data including news headlines and COVID-19 related tweets. Several future directions include performing such comparisons between matrix and tensor-based methods on different types of data (e.g. hyperspectral image data) for topic modeling and other applications.

Bibliography

[1] Miju Ahn, Nicole Eikmeier, Jamie Haddock, Lara Kassab, Alona Kryshchenko, Kathryn Leonard, Deanna Needell, RWMA Madushani, Elena Sizikova, and Chuntian Wang. On large-scale dynamic topic modeling with nonnegative CP tensor decomposition. *arXiv preprint arXiv:2001.00631*, 2020.

[2] Animashree Anandkumar, Daniel Hsu, and Sham M Kakade. A method of moments for mixture models and hidden Markov models. In *Conference on Learning Theory*, pages 33–1. JMLR Workshop and Conference Proceedings, 2012.

[3] Pia Anttila, Pentti Paatero, Unto Tapper, and Olli Järvinen. Source identification of bulk wet deposition in Finland by positive matrix factorization. *Atmospheric Environment*, 29(14):1705–1718, 1995.

[4] Brett W Bader, Michael W Berry, and Murray Browne. Discussion tracking in Enron email using PARAFAC. In *Survey of Text Mining II*, pages 147–163. Springer, 2008.

[5] Sanaz Bahargam and Evangelos Papalexakis. A constrained coupled matrix-tensor factorization for learning time-evolving and emerging topics. *arXiv preprint arXiv:1807.00122*, 2018.

[6] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proc. Int. Conf. Mach. Learn.* Citeseer, 2002.

[7] BBC. Li Wenliang: Coronavirus kills Chinese whistleblower doctor, 2020.

[8] Christian F Beckmann and Stephen M Smith. Tensorial extensions of independent component analysis for multisubject FMRI analysis. *Neuroimage*, 25(1):294–311, 2005.

[9] Mark Belford, Brian MacNamee, and Derek Greene. Ensemble topic modeling via matrix factorization. In *24th Irish Conference on Artificial Intelligence and Cognitive Science*

(AICS'16), Dublin, Ireland, 20-21 September 2016, volume 1751. CEUR Workshop Proceedings, 2016.

[10] Robert M Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *SiGKDD Explorations*, 9(2):75–79, 2007.

[11] James Bennett and Stan Lanning. The Netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007, page 35. New York, NY, USA., 2007.

[12] M. W. Berry and M. Browne. Email surveillance using non-negative matrix factorization. *Comput. Math. Organ. Th.*, 11(3):249–264, 2005.

[13] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, 1999.

[14] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. " O'Reilly Media, Inc.", 2009.

[15] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

[16] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, 2006.

[17] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003.

[18] Svenja Boberg, Thorsten Quandt, Tim Schatto-Eckrodt, and Lena Frischlich. Pandemic populism: Facebook pages of alternative news media and the Corona crisis–a computational content analysis. *arXiv preprint arXiv:2004.02566*, 2020.

[19] Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern recognition*, 41(4):1350–1362, 2008.

[20] Ioan Buciu. Non-negative matrix factorization, a new tool for feature extraction: Theory and applications. *International Journal of Computers, Communications and Control*, 3(3):67–74, 2008.

[21] 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.

[22] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[23] Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[24] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[25] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[26] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[27] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[28] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Comput. Intel. Neurosc.*, 2009, 2008.

[29] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[30] Emily Chen, Kristina Lerman, and Emilio Ferrara. Tracking social media discourse about the COVID-19 pandemic: Development of a public Coronavirus Twitter data set. *JMIR Public Health and Surveillance*, 6(2):e19273, 2020.

[31] Huiyuan Chen and Jing Li. Modeling relational drug-target-disease interactions via tensor factorization with multiple web sources. In *The World Wide Web Conference*, pages 218–227, 2019.

[32] Liangzhe Chen, KSM Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B Aditya Prakash. Flu gone viral: Syndromic surveillance of flu on Twitter using temporal topic models. In *2014 IEEE International Conference on Data Mining*, pages 755–760. IEEE, 2014.

[33] Y. Chen, M. Rege, M. Dong, and J. Hua. Non-negative matrix factorization for semi-supervised data clustering. *Knowl. Inf. Syst.*, 17(3):355–379, 2008.

[34] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. In *International Conference on Machine Learning*, pages 674–682. PMLR, 2014.

[35] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Completing any low-rank matrix, provably. *The Journal of Machine Learning Research*, 16(1):2999–3034, 2015.

[36] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

[37] Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

[38] Y. Cho and L. K. Saul. Nonnegative matrix factorization for semi-supervised dimensionality reduction. *arXiv preprint arXiv:1112.3714*, 2011.

[39] Despoina Christou. Feature extraction using latent dirichlet allocation and neural networks: a case study on movie synopses. *arXiv preprint arXiv:1604.01272*, 2016.

[40] A. Cichocki, R. Zdunek, and S. Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *Proc. Int. Conf. Acoust. Spe. Sig. Process.*, volume 5, pages V–V. IEEE, 2006.

[41] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[42] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Nonnegative matrix and tensor factorization [lecture notes]. *IEEE signal processing magazine*, 25(1):142–145, 2007.

[43] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

[44] R. de Fréin, K. Drakakis, S. Rickard, and A. Cichocki. Analysis of financial data using non-negative matrix factorization. In *Proc. Int. Mathematical Forum*, volume 3(38), pages 1853–1870. Hikari, 2008.

[45] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data An.*, 52(8):3913–3927, 2008.

[46] David L Donoho and Philip B Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.

[47] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):1–27, 2011.

[48] P. Favaro and S. Soatto. *3-d shape estimation and image restoration: Exploiting defocus and motion-blur.* Springer Science & Business Media, 2007.

[49] Maryam Fazel. *Matrix rank minimization with applications.* PhD thesis, Stanford University, 2002.

[50] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the 2003 American Control Conference, 2003*, volume 3, pages 2156–2162. IEEE, 2003.

[51] W. Fei, L. Tao, and Z. Changshui. Semi-supervised clustering via matrix factorization. In *Proc. SIAM Int. Conf. on Data Mining*, 2008.

[52] Emilio Ferrara. What types of COVID-19 conspiracies are populated by Twitter bots? *First Monday*, 25(6), May 2020.

[53] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[54] Massimo Fornasier, Holger Rauhut, and Rachel Ward. Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM Journal on Optimization*, 21(4):1614–1640, 2011.

[55] Eibe Frank and Remco R Bouckaert. Naive Bayes for text classification with unbalanced classes. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 503–510. Springer, 2006.

[56] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *J. Mach. Learn. Res.*, 5(Jan):73–99, 2004.

[57] E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *Proc. ACM SIGIR Conf. on Research and Development in Inform. Retrieval*, pages 601–602, 2005.

[58] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, et al. An exemplar-based NMF approach to audio event detection. In *Proc. IEEE Workshop on Appl. Sig. Process. to Audio and Acoust.*, pages 1–4. IEEE, 2013.

[59] Donald Goldfarb and Shiqian Ma. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics*, 11(2):183–210, 2011.

[60] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[61] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

[62] Jon Green, Jared Edgerton, Daniel Naftel, Kelsey Shoub, and Skyler J Cranmer. Elusive consensus: Polarization in elite communication on the COVID-19 pandemic. *Science Advances*, 6(28):eabc2717, 2020.

[63] Derek Greene and James P Cross. Exploring the political agenda of the European parliament using a dynamic topic modeling approach. *Political Analysis*, 25(1):77–94, 2017.

[64] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[65] David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical review letters*, 105(15):150401, 2010.

[66] D. Guillamet and J. Vitria. Non-negative matrix factorization for face recognition. In *Proc. Catalonian Conf. on Artif. Intel.*, pages 336–344. Springer, 2002.

[67] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(Mar):1157–1182, 2003.

[68] Jamie Haddock, Lara Kassab, and Sixian Li. SSNMF code. https://pypi.org/project/ssnmf/.

[69] Jamie Haddock, Lara Kassab, Sixian Li, Alona Kryshchenko, Rachel Grotheer, Elena Sizikova, Chuntian Wang, Thomas Merkh, RWMA Madushani, Miju Ahn, et al. Semi-supervised nonnegative matrix factorization for document classification. *arXiv preprint arXiv:2010.07956*, 2020.

[70] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[71] Richard A Harshman et al. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. 1970.

[72] Samuel W Hasinoff. Photon, Poisson noise. 2014.

[73] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[74] Le Thi Khanh Hien and Nicolas Gillis. Algorithms for nonnegative matrix factorization with the Kullback–Leibler divergence. *Journal of Scientific Computing*, 87(3):1–32, 2021.

[75] Geoffrey E Hinton and Russ R Salakhutdinov. Replicated softmax: an undirected topic model. *Advances in neural information processing systems*, 22:1607–1614, 2009.

[76] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent Dirichlet allocation. In *advances in neural information processing systems*, pages 856–864. Citeseer, 2010.

[77] Kate Holland, R Warwick Blood, Michelle Imison, Simon Chapman, and Andrea Fogarty. Risk, expert uncertainty, and Australian news media: public and private faces of expert opinion during the 2009 swine flu pandemic. *Journal of Risk Research*, 15(6):657–671, 2012.

[78] Seyedmehdi Hosseinimotlagh and Evangelos E Papalexakis. Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. In *Proceedings of the Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*, 2018.

[79] P. O. Hoyer. Non-negative sparse coding. In *Proc. IEEE Workshop on Neural Networks for Sig. Process.*, pages 557–565. IEEE, 2002.

[80] Jiajun Hu, Xiaobing Sun, David Lo, and Bin Li. Modeling the evolution of development topics using dynamic topic models. In *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 3–12. IEEE, 2015.

[81] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 663–672, 2010.

[82] Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

[83] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

[84] Y. Jia, S. Kwong, J. Hou, and W. Wu. Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization. *IEEE T. Neur. Net. Lear.*, 2019.

[85] Y. Jia, Y. Wang, C. Turk, and M. Hu. Fisher non-negative matrix factorization for learning local features. In *Proc. Asian Conf. Comp. Vis.*, pages 27–30. Citeseer, 2004.

[86] K Kakimoto, H Kamiya, T Yamagishi, T Matsui, M Suzuki, and T Wakita. Initial investigation of transmission of COVID-19 among crew members during quarantine of a cruise ship — Yokohama, Japan, February 2020. *MMWR Morb Mortal Wkly Rep*, 69:312–313, 2020.

[87] Lara Kassab, Henry Adams, and Deanna Needell. Structured IRLS code. https://github.com/lara-kassab/structured-matrix-completion-IRLS.

[88] Lara Kassab, Henry Adams, and Deanna Needell. An adaptation for iterative structured matrix completion. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 1451–1456, 2020. Journal Version to appear in *Foundations of Data Science*, 2021.

[89] Lara Kassab, Alona Kryshchenko, Hanbaek Lyu, Denali Molitor, Deanna Needell, and Elizaveta Rebrova. Detecting short-lasting topics using nonnegative tensor decomposition. *arXiv preprint arXiv:2010.01600*, 2020.

[90] Raghunandan H Keshavan and Sewoong Oh. A gradient descent algorithm on the Grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*, 2009.

[91] D. Kim, S. Sra, and I. S. Dhillon. Fast projection-based methods for the least squares nonnegative matrix approximation problem. *Stat. Anal. Data Min.*, 1(1):38–51, 2008.

[92] H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM J. Matrix Anal. A.*, 30(2):713–730, 2008.

[93] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. Technical report, Stanford, 2002.

[94] Bennett Kleinberg, Isabelle van der Vegt, and Maximilian Mozes. Measuring emotions in the COVID-19 real world worry dataset. *arXiv preprint arXiv:2004.04225*, 2020.

[95] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[96] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[97] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *Journal of Machine Learning Research*, 20(26):1–6, 2019.

[98] Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*, pages 215–243. Springer, 2015.

[99] Rohit Kulkarni. A Million News Headlines, 2018.

[100] Christian Kümmerle and Juliane Sigl. Harmonic mean iteratively reweighted least squares for low-rank matrix recovery. *The Journal of Machine Learning Research*, 19(1):1815–1863, 2018.

[101] William H Lawton and Edward A Sylvestre. Self modeling curve resolution. *Technometrics*, 13(3):617–633, 1971.

[102] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. Adv. Neur. In.*, pages 556–562, 2001.

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

[104] H. Lee, J. Yoo, and S. Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Proc. Let.*, 17(1):4–7, 2009.

[105] Kiryung Lee and Yoram Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. *IEEE Transactions on Information Theory*, 56(9):4402–4416, 2010.

[106] L. Li, G. Lebanon, and H. Park. Fast Bregman divergence NMF using Taylor expansion and coordinate descent. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 307–315, 2012.

[107] C. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19(10):2756–2779, 2007.

[108] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[109] J. Liu, D. Wang, Y. Gao, C. Zheng, Y. Xu, and J. Yu. Regularized non-negative matrix factorization for identifying differentially expressed genes and clustering samples: a survey. *IEEE/ACM T. Comput. Bio. Bioin.*, 15(3):974–987, 2017.

[110] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[111] Hanbaek Lyu. Convergence of block coordinate descent with diminishing radius for nonconvex optimization. *arXiv preprint arXiv:2012.03503*, 2020.

[112] Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

[113] C. D. Manning, H. Schütze, and P. Raghavan. *Introduction to information retrieval.* Cambridge university press, 2008.

[114] Chris Manning. *I. Introduction.* ISEAS Publishing, 1988.

[115] Karthik Mohan and Maryam Fazel. IRLS code. https://faculty.washington.edu/mfazel/. [Online; accessed 01-Aug-2019].

[116] Karthik Mohan and Maryam Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proceedings of the 2010 American Control Conference*, pages 2953–2959. IEEE, 2010.

[117] Karthik Mohan and Maryam Fazel. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13(Nov):3441–3473, 2012.

[118] Denali Molitor and Deanna Needell. Matrix completion for structured observations. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE, 2018.

[119] NBCNews. Trump calls Coronavirus democrats' 'new hoax', 2020.

[120] NBCNews. Washington state man becomes first U.S. death from Coronavirus, 2020.

[121] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[122] Duy Khuong Nguyen and Tu Bao Ho. Fast parallel randomized algorithm for nonnegative matrix factorization with KL divergence for large sparse datasets. *arXiv preprint arXiv:1604.04026*, 2016.

[123] Catherine Ordun, Sanjay Purushotham, and Edward Raff. Exploratory analysis of COVID-19 tweets using topic modeling, umap, and digraphs. *arXiv preprint arXiv:2005.03082*, 2020.

[124] 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.

[125] Evangelos Papalexakis, Konstantinos Pelechrinis, and Christos Faloutsos. Spotting misbehaviors in location-based social networks using tensors. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 551–552, 2014.

[126] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Adaptive model for dynamic and temporal topic modeling from big data using deep learning architecture. *International Journal of Intelligent Systems and Applications*, 11(6):13, 2019.

[127] V. Pauca, F. Shahnaz, M. Berry, and R. Plemmons. Text mining using non-negative matrix factorizations. In *Proc. SIAM Int. Conf. on Data Mining*, pages 452–456. SIAM, 2004.

[128] Michael J Paul and Mark Dredze. You are what you tweet: Analyzing Twitter for public health. In *Fifth international AAAI conference on weblogs and social media*. Citeseer, 2011.

[129] Michael J Paul and Mark Dredze. Discovering health topics in social media using topic models. *PloS one*, 9(8):e103408, 2014.

[130] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[131] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[132] Politico. Trump puts Pence in charge of Coronavirus response, 2020.

[133] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

[134] Stephan Rabanser, Oleksandr Shchur, and Stephan Günnemann. Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*, 2017.

[135] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[136] C Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

[137] Ilya Razenshteyn, Zhao Song, and David P Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 250–263, 2016.

[138] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

[139] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[140] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[141] J. Rennie. 20 Newsgroups, 2008.

[142] Geneviève Robin, Olga Klopp, Julie Josse, Éric Moulines, and Robert Tibshirani. Main effects and interactions in mixed and incomplete data frames. *Journal of the American Statistical Association*, pages 1–12, 2019.

[143] German Rodriguez. Generalized linear models. *Lecture notes [acessado em 1 Mai 2010]. Disponível em: http://data. princeton. edu/wws509/notes*, 2001.

[144] Athanasios A Rontogiannis, Paris V Giampouras, and Konstantinos D Koutroumbas. Online reweighted least squares robust PCA. *IEEE Signal Processing Letters*, 27:1340–1344, 2020.

[145] 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, 2012.

[146] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[147] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352*, 2016.

[148] Hao Sha, Mohammad Al Hasan, George Mohler, and P Jeffrey Brantingham. Dynamic topic modeling of the COVID-19 Twitter narrative among us governors and cabinet executives. *arXiv preprint arXiv:2004.11692*, 2020.

[149] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons. Document clustering using nonnegative matrix factorization. *Inform. Process. Manag.*, 42(2):373–386, 2006.

[150] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799, 2005.

[151] R. Sheikhpour, M. Sarram, S. Gharaghani, and M. Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recogn.*, 64:141–158, 2017.

[152] Amit Singer. A remark on global positioning from local distances. *Proceedings of the National Academy of Sciences*, 105(28):9507–9511, 2008.

[153] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

[154] Aude Sportisse, Claire Boyer, and Julie Josse. Imputation and low-rank estimation with missing non at random data. *arXiv preprint arXiv:1812.11409*, 2018.

[155] S. Sra and I. S. Dhillon. Generalized nonnegative matrix approximations with Bregman divergences. In *Proc. Adv. Neur. In.*, pages 283–290, 2006.

[156] Statista. Number of new Coronavirus (COVID-19) cases in South Korea from January 20 to August 21, 2020, 2020.

[157] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640):15, 2010.

[158] TwitterIR. Q2' 2020 shareholder letter. https://s22.q4cdn.com/826641620/files/doc_financials/2020/q2/Q2-2020-Shareholder-Letter.pdf, July 2020.

[159] Isabelle Van der Vegt and Bennett Kleinberg. Women worry about family, men about the economy: Gender differences in emotional responses to COVID-19. *arXiv preprint arXiv:2004.08202*, 2020.

[160] T. Virtanen, A. T. Cemgil, and S. Godsill. Bayesian extensions to non-negative matrix factorisation for audio signal modelling. In *Proc. IEEE Int. Conf. on Acoust., Speech and Sig. Process.*, pages 1825–1828. IEEE, 2008.

[161] World Vision. 2004 Indian ocean earthquake and tsunami: Facts, faqs, and how to help.

[162] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proc. Int. Conf. Mach. Learn.*, volume 1, pages 577–584, 2001.

[163] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

[164] W. Wang and M. A. Carreira-Perpinán. The role of dimensionality reduction in classification. In *Proc. AAAI Conf. on Artif. Intel.*, pages 2128–2134, 2014.

[165] Wire. 'No meat, no Coronavirus' makes no sense, 2020.

[166] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[167] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM, 2010.

[168] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[169] Y. Xue, C. S. Tong, W. Chen, W. Zhang, and Z. He. A modified non-negative matrix factorization algorithm for face recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 495–498. IEEE, 2006.

[170] Kai-Cheng Yang, Christopher Torres-Lugo, and Filippo Menczer. Prevalence of low-credibility information on Twitter during the COVID-19 outbreak. *arXiv preprint arXiv:2004.14484*, 2020.

[171] Z. Yang, H. Zhang, Z. Yuan, and E. Oja. Kullback-Leibler divergence for nonnegative matrix factorization. In *Proc. Int. Conf. on Artif. Neural Networks*, pages 250–257. Springer, 2011.

[172] Hui Yin, Shuiqiao Yang, and Jianxin Li. Detecting topic and sentiment dynamics due to COVID-19 pandemic using social media. *arXiv preprint arXiv:2007.02304*, 2020.

[173] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas. Exploiting discriminant information in non-negative matrix factorization with application to frontal face verification. *IEEE T. Neural Networ.*, 17(3):683–695, 2006.

[174] Juan Zhao, Yun Zhang, David J Schlueter, Patrick Wu, Vern Eric Kerchberger, S Trent Rosenbloom, Quinn S Wells, QiPing Feng, Joshua C Denny, and Wei-Qi Wei. Detect-

ing time-evolving phenotypic topics via tensor factorization on electronic health records: Cardiovascular disease case study. *Journal of biomedical informatics*, 98:103270, 2019.

[175] Caleb Ziems, Bing He, Sandeep Soni, and Srijan Kumar. Racism is a virus: Anti-asian hate and counterhate in social media during the COVID-19 crisis, 2020.

Appendix A

Additional SSNMF Models Derivations and Results

We provide in Appendix A.1 the remaining maximum likelihood estimation derivations (1, 3, and 4) from Section 3.2.2, and additional experimental results on the 20 Newsgroups dataset in Appendices A.2 and A.3.

A.1 Maximum Likelihood Estimation Derivations

First, we demonstrate that the MLE, in the case that the uncertainty on the **X** and **Y** observations is Gaussian distributed, is a specific instance of $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF of [104]. Our models for the distribution of the observed entries of **X** and **Y** will assume that the mean is given by an exact factorization, $\mathbb{E}[\mathbf{X}] = \mathbf{AS}$ and $\mathbb{E}[\mathbf{Y}] = \mathbf{BS}$, and the uncertainty in each set of observations is governed by a Gaussian distribution. That is, we consider the hierarchical models for **X** and **Y** in which

$$\mathbf{X}_{\gamma,\tau} = \sum_{i=1}^{r} x_{\gamma,i,\tau} \text{ and } x_{\gamma,i,\tau} \sim \mathcal{N}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, \sigma_1\right),$$

$$\mathbf{Y}_{\eta,\tau} = \sum_{i=1}^{r} y_{\eta,i,\tau} \text{ and } y_{\eta,i,\tau} \sim \mathcal{N}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, \sigma_2\right).$$

Here and throughout, γ and η are row indices of **X** and **Y** respectively, τ is a column index of **X** and **Y**, and i indexes the random variable summands which form $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$. Note then that

$$\mathbf{X}_{\gamma,\tau} \sim \mathcal{N}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, r\sigma_1\right) \text{ and } \mathbf{Y}_{\eta,\tau} \sim \mathcal{N}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, r\sigma_2\right)$$

due to the summable property of Gaussian random variables. We note that this assumes different Gaussian models of uncertainty on the two collections of rows of the NMF (3.4).

Assuming that the set of $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$ are statistically independent conditional on **A**, **B**, and **S**, we have that the likelihood

$$p(\mathbf{X},\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{S}) = \prod_{\gamma,\tau} \mathcal{N}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}, r\sigma_1\right) \prod_{\eta,\tau} \mathcal{N}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, r\sigma_2\right). \tag{A.1}$$

We apply the monotonic natural logarithmic function to the likelihood, and ignore terms that do not vary with the factor matrices. This transforms the likelihood function into a $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF objective, while preserving the maximizer. That is, the log likelihood (excluding additive terms which are constant with respect to **A**, **B**, and **S**) is

$$\begin{aligned} \ln p\,(\mathbf{X},\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{S}) &=^{+} -\frac{1}{2r\sigma_1}\sum_{\gamma,\tau}\left(\mathbf{X}_{\gamma,\tau} - \sum_{i=1}^{r}\mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right)^2 - \frac{\lambda}{2r\sigma_2}\sum_{\eta,\tau}\left(\mathbf{Y}_{\eta,\tau} - \sum_{i=1}^{r}\mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right)^2 \\ &=^{+} -\frac{1}{2r\sigma_1}\left[\|\mathbf{X}-\mathbf{AS}\|_F^2 + \frac{\sigma_1}{\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2\right]. \end{aligned}$$

Thus, the maximum likelihood estimators for **A**, **B**, and **S** are given by

$$\underset{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{AS}\|_F^2 + \frac{\sigma_1}{\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2.$$

We see that the MLE in the case of Gaussian uncertainty on both sets of observations, **X** and **Y**, is a specific instance of $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF objective where the regularization parameter λ, which defines the relative weighting of the supervision term, is given as a ratio of the variances of the distributions.

Next, we demonstrate that the MLE, in the case that the uncertainty on **X** is Poisson distributed and on **Y** is Gaussian distributed, is a specific instane of the $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF model. This MLE derivation follows from that of 2 by swapping the roles of **X** and **Y**, and rescaling the resulting log likelihood; however, we include a sketch of the derivation to be thorough.

Again, our models for observed **X** and **Y** assume that the mean is given by an exact factorization, $\mathbb{E}[\mathbf{X}] = \mathbf{AS}$ and $\mathbb{E}[\mathbf{Y}] = \mathbf{BS}$, with the uncertainty in **X** governed by a Poisson distribution and the uncertainty in **Y** governed by a Gaussian distribution. That is, we consider the hierarchical models for **X** and **Y** in which

$$\mathbf{X}_{\gamma,\tau} = \sum_{i=1}^{r} x_{\gamma,i,\tau} \text{ and } x_{\gamma,i,\tau} \sim \mathscr{PO}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right),$$

$$\mathbf{Y}_{\eta,\tau} = \sum_{i=1}^{r} y_{\eta,i,\tau} \text{ and } y_{\eta,i,\tau} \sim \mathscr{N}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, \sigma_2\right).$$

Note then that

$$\mathbf{X}_{\gamma,\tau} \sim \mathscr{PO}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right) \text{ and } \mathbf{Y}_{\eta,\tau} \sim \mathscr{N}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}, r\sigma_2\right)$$

due to the summable property of Gaussian and Poisson random variables. We note this assumes a Poisson and Gaussian model of uncertainty on the two collections of rows of the NMF (3.4).

Then proceeding as in (A.1) and (3.6) and assuming that the set of $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$ are statistically independent conditional on **A**, **B**, and **S**, we have that the log likelihood (excluding additive terms which are constant with respect to **A**, **B**, and **S**) is

$$\ln p\left(\mathbf{X},\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{S}\right) =^{+} -\left[D(\mathbf{X}\|\mathbf{AS}) + \frac{1}{2r\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2\right].$$

Thus, the maximum likelihood estimators for **A**, **B**, and **S** are given by

$$\operatorname*{argmin}_{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0} D(\mathbf{X}\|\mathbf{AS}) + \frac{1}{2r\sigma_2}\|\mathbf{Y}-\mathbf{BS}\|_F^2.$$

We see that the MLE in the case of Poisson uncertainty on the observations in **X** and Gaussian uncertainty on the observations in **Y** is a specific instance of the $(D(\cdot\|\cdot), \|\cdot\|_F)$-SSNMF objective where the regularization parameter λ is the inverse of a multiple of the variance of the Gaussian distribution.

Finally, we demonstrate that the MLE, in the case that the uncertainty on **X** and **Y** are Poisson distributed, is a specific instance of the $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF model. This result follows from [28, 48, 160]; we sketch the derivation to be thorough.

Again, we assume that the distributions of the observed **X** and Y have means given by an exact factorization, $\mathbb{E}[\mathbf{X}] = \mathbf{AS}$ and $\mathbb{E}[\mathbf{Y}] = \mathbf{BS}$, with the uncertainty in both governed by a Poisson distribution. That is, we consider the hierarchical models for **X** and **Y** in which

$$\mathbf{X}_{\gamma,\tau} = \sum_{i=1}^{r} x_{\gamma,i,\tau} \text{ and } x_{\gamma,i,\tau} \sim \mathscr{PO}\left(x_{\gamma,i,\tau} \middle| \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right),$$

$$\mathbf{Y}_{\eta,\tau} = \sum_{i=1}^{r} y_{\eta,i,\tau} \text{ and } y_{\eta,i,\tau} \sim \mathscr{PO}\left(y_{\eta,i,\tau} \middle| \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right).$$

Note then that

$$\mathbf{X}_{\gamma,\tau} \sim \mathscr{PO}\left(\mathbf{X}_{\gamma,\tau} \middle| \sum_{i=1}^{r} \mathbf{A}_{\gamma,i}\mathbf{S}_{i,\tau}\right) \text{ and } \mathbf{Y}_{\eta,\tau} \sim \mathscr{PO}\left(\mathbf{Y}_{\eta,\tau} \middle| \sum_{i=1}^{r} \mathbf{B}_{\eta,i}\mathbf{S}_{i,\tau}\right)$$

due to the summable property of Poisson random variables. We note that assumes different Poisson models of uncertainty on the two collections of rows of the NMF (3.4).

Then proceeding as in (A.1) and (3.6) and assuming that the set of $\mathbf{X}_{\gamma,\tau}$ and $\mathbf{Y}_{\eta,\tau}$ are statistically independent conditional on **A**, **B**, and **S**, we have that the log likelihood (excluding additive terms which are constant with respect to **A**, **B**, and **S**) is

$$\ln p\left(\mathbf{X},\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{S}\right) =^{+} -\left[D(\mathbf{X}\|\mathbf{AS}) + D(\mathbf{Y}\|\mathbf{BS})\right].$$

Thus, the maximum likelihood estimators for **A**, **B**, and **S** are given by

$$\operatorname*{argmin}_{\mathbf{A},\mathbf{B},\mathbf{S}\geq 0} D(\mathbf{X}\|\mathbf{AS}) + D(\mathbf{Y}\|\mathbf{BS}).$$

We see that the MLE in the case of Poisson uncertainty on the observations in **X** and **Y** is a specific instance of the $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF objective where the regularization parameter is $\lambda = 1$.

A.2 Results of SSNMF Models on 20 Newsgroups Dataset

In this section, we include additional analysis and results for the SSNMF models on 20 Newsgroups dataset. First, we summarize in Table A.1 the hyperparameters used for the methods described in Section 3.3. We select the hyperparameters that result in the highest average classification accuracy of the validation set. For the SSNMF models, we search over $tol \in \{10^{-4}, 10^{-3}, 10^{-2}\}$, and $\lambda \in \{10, 10^2, 10^3\}$, and for the NMF model, we search over $tol \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$.

Table A.1: Hyperparameter selection for NMF and SSNMF models by selecting the hyperparameters that result with the highest average classification accuracy of the validation set (over 10 trials).

Model	hyperparameters
$(\|\cdot\|_F, \|\cdot\|_F)$	$tol = 10^{-4}$, $\lambda = 10^2$
$(\|\cdot\|_F, D(\cdot\|\cdot))$	$tol = 10^{-4}$, $\lambda = 10$
$(D(\cdot\|\cdot), \|\cdot\|_F)$	$tol = 10^{-3}$, $\lambda = 10^2$
$(D(\cdot\|\cdot), D(\cdot\|\cdot))$	$tol = 10^{-3}$, $\lambda = 10^3$
$\|\cdot\|_F$-NMF	$tol = 10^{-4}$
$D(\cdot\|\cdot)$-NMF	$tol = 10^{-5}$

As in Section 3.3, we consider the “typical" (achieving median accuracy within trials) decomposition for the NMF models, and the remaining SSNMF models. We display in Figures A.1-A.3 the $\mathbf{B}_{\text{train}}$ matrices for each of the median accuracy SSNMF decompositions, and in Figures A.4 and A.5 the coefficients matrix of the SVM classifier for the median accuracy for each of $\|\cdot\|_F$-NMF and $D(\cdot\|\cdot)$-NMF decompositions, respectively. Further, we report in Tables A.2-A.6 the top 10 keywords representing each topic for each of the models.

For the $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF model, we (qualitatively) observe from Table A.2 that topics 4, 5, 7 and 10 are overlapping topics associated to the class Computers; see Figure A.1. Similarly, topics 1 and 9 that capture the subjects of crypt(ography), electronics, and space, have various overlapping keywords (“space", “key",“chip"), and are associated with the class Sciences. On the other hand, topics associated to class Politics and Religion are less overlapping. Lastly, top-

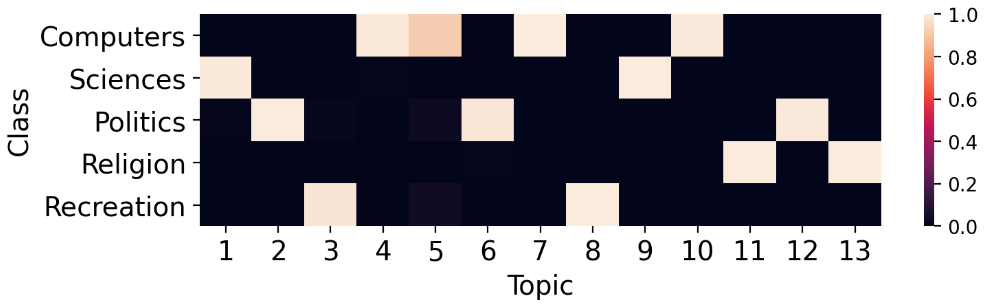

Figure A.1: The normalized $\mathbf{B}_{\text{train}}$ matrix for $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF decomposition corresponding to the median test classification accuracy equal to 79.44%. Each column is normalized to represent the distribution of the topic over classes.

Table A.2: Top keywords representing each topic of the $(\|\cdot\|_F, \|\cdot\|_F)$-SSNMF model referred to in Figure A.1.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Keywords	would	people	game	use	x	would	x	game	would	please	god	israel	god
	like	israel	team	thanks	software	jews	thanks	car	key	x	church	people	one
	space	gun	car	x	c	fbi	get	team	could	would	jesus	would	would
	one	one	year	using	know	time	need	like	one	use	one	one	people
	chip	jews	hockey	window	r	israel	image	games	space	anyone	would	killed	jesus
	key	would	espn	know	thanks	like	window	baseball	use	like	people	armenians	believe
	use	armenian	would	graphics	widget	law	problem	one	chip	graphics	think	police	christ
	good	government	players	pc	system	arabs	windows	think	know	work	like	jewish	religion
	phone	said	nhl	program	please	government	mac	would	get	help	say	well	think
	edu	turkish	games	anyone	motif	right	version	get	like	apple	faith	israeli	bible
Hard	electronics	mideast	hockey	graphics	windows	guns	windows	baseball	crypt	graphics	christian	guns	christian
Score	0.2060	0.6594	0.3411	0.2211	0.1047	0.1466	0.5698	0.7500	0.7641	0.1441	0.5226	0.1625	0.4574
Soft	space	mideast	hockey	graphics	windows	guns	windows	baseball	crypt	graphics	christian	mideast	christian
Score	0.3135	0.4560	0.4222	0.2389	0.1951	0.2278	0.3564	0.5933	0.6276	0.2128	0.4983	0.2480	0.4525

ics 3 and 8 are recreation topics (“game", “team", “car") relating to autos where in addition topic 3 (“hockey", “player", “nhl") is specific to hockey and topic 8 (“baseball") is specific to baseball.

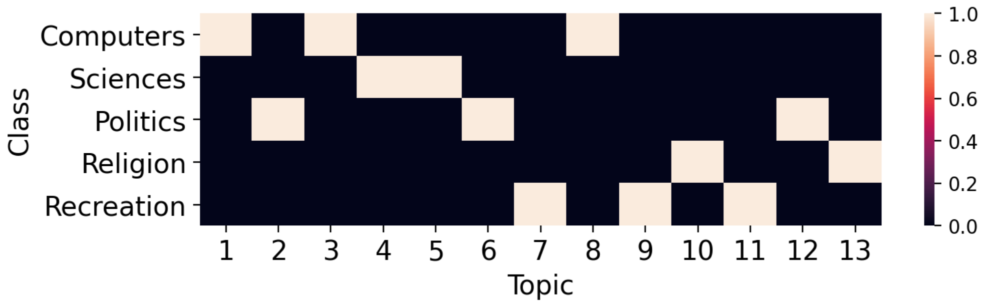

Figure A.2: The normalized $\mathbf{B}_{\text{train}}$ matrix for $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF decomposition corresponding to the median test classification accuracy equal to 79.56%. Each column is normalized to represent the distribution of the topic over classes.

Table A.3: Top keywords representing each topic of the $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF model referred to in Figure A.2.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Keywords	x	israel	thanks	would	would	people	game	ordinary	car	god	game	people	god
	know	would	x	chip	one	would	player	yeah	team	jesus	good	gun	one
	would	people	anyone	use	key	jews	hockey	monitors	game	deleted	one	one	would
	use	government	get	clipper	could	israel	espn	ok	games	science	car	guns	people
	window	israeli	mac	space	space	gun	would	big	like	would	get	israel	jesus
	graphics	one	sun	government	like	fbi	baseball	know	year	moses	anyone	said	church
	please	fbi	graphics	key	know	one	new	shareware	get	post	great	armenian	think
	software	armenians	file	much	get	law	wings	way	would	passages	play	government	bible
	windows	armenian	one	get	use	fire	think	anyone	one	come	year	well	believe
	mac	also	please	people	chip	think	players	good	think	commandments	better	would	christian
Hard	windows	mideast	graphics	crypt	electronics	guns	hockey	graphics	autos	atheism	baseball	mideast	christian
Score	0.6429	0.3548	0.3719	0.3272	0.6478	0.2784	0.2268	0.0067	0.7676	0.0463	0.0405	0.3865	0.9298
Soft	windows	mideast	windows	crypt	electronics	guns	hockey	graphics	autos	christian	baseball	mideast	christian
Score	0.5262	0.3315	0.3821	0.3973	0.5444	0.2988	0.2717	0.0659	0.5669	0.1296	0.1274	0.3363	0.8198

For the $(\|\cdot\|_F, D(\cdot\|\cdot))$-SSNMF model, we (qualitatively) observe from Table A.3 that topic 2 (“armenian"), topic 6 (“jews",“gun", “fire") and topic 12 (“guns") are related political topics (“people", “government", “israel", “fbi"). The topics are associated to the class Politics; see Figure A.2. Further, recreation topics include topic 7 (“player", “hockey", “baseball") relating to hockey and baseball, topic 9 (“car",“team") relating to autos, and a broad topic 11 (“game", “good", “great", “play", “better"). Lastly, topics 1, 3 and 8 are associated to class Computers. Topic 1 (“window",“graphics",“software",“windows"), and topic 3 (“mac", “sun", “graphics"), are specific in comparison to topic 8 (“ordinary",“yeah",“monitor") which is broad and not as informative.

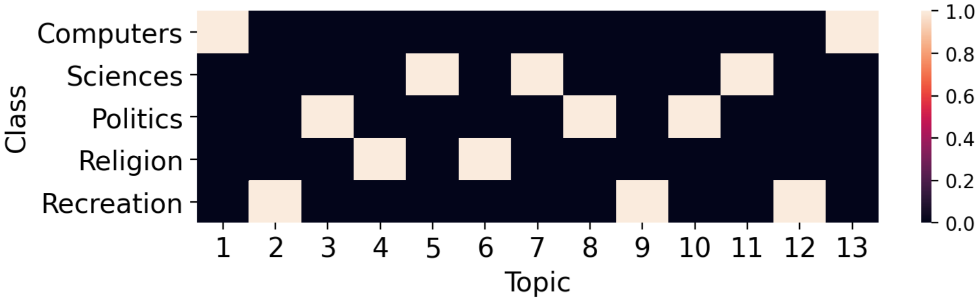

Figure A.3: The normalized $\mathbf{B}_{\text{train}}$ matrix for $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF decomposition corresponding to the median test classification accuracy equal to 81.39%. Each column is normalized to represent the distribution of the topic over classes.

For the $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF model, we observe from Figure A.3 that topic 5 (“space", “moon", “time"), topic 7 (“key",“chip",“clipper"), and topic 11 (“data",“government",“buy") are

Table A.4: Top keywords representing each topic of the $(D(\cdot\|\cdot), D(\cdot\|\cdot))$-SSNMF model referred to in Figure A.3.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Keywords	thanks	game	israel	god	would	god	key	would	game	people	one	car	x
	x	games	one	would	space	atheists	would	people	win	gun	like	team	anyone
	mac	would	arab	one	could	one	chip	government	etc	get	use	game	thanks
	know	think	people	jesus	u	people	clipper	gun	turbo	right	get	year	graphics
	would	get	government	church	use	would	could	said	know	jews	would	like	get
	problem	back	would	people	moon	think	space	israel	games	armenian	think	hockey	use
	please	hit	fire	believe	time	paul	know	one	get	well	data	players	window
	use	well	well	bible	old	jesus	one	jews	would	us	anyone	last	would
	one	one	like	think	one	know	using	armenians	cup	time	government	one	please
	se	like	israeli	christian	may	also	like	batf	find	armenians	buy	baseball	know
Hard	graphics	hockey	mideast	christian	crypt	atheism	crypt	guns	autos	mideast	electronics	baseball	windows
Score	0.2982	0.0993	0.4487	0.8947	0.1246	0.0876	0.5897	0.1875	0.0545	0.3526	0.3256	0.8521	0.7475
Soft	graphics	hockey	mideast	christian	crypt	christian	crypt	mideast	hockey	mideast	space	baseball	windows
Score	0.3682	0.1746	0.3643	0.7596	0.2175	0.1989	0.4307	0.2602	0.1306	0.3252	0.3199	0.6653	0.6240

associated with class Sciences. Further, we (qualitatively) observe from Table A.4 that topic 4 (“church", “believe",“bible"), and topic 6 (“atheists", “paul") are both related to religion (“god", “jesus") and are associated to class Religion; see Figure A.3. Lastly, topic 12 (“car", “hockey", “players", “baseball") is a recreation topic that captures autos, hockey, and baseball subjects, whereas topics 2 and 9 are broad and not as informative.

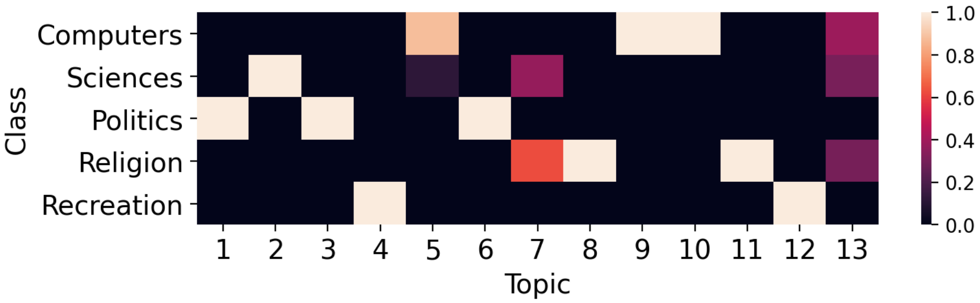

Figure A.4: The coefficients matrix of the SVM classifier (with NMF-$\|\cdot\|_F$) corresponding to the median test classification accuracy equal to 71.67%. Here, all negative coefficients are thresholded to 0, and then each column is normalized to showcases the distribution of the topic over classes.

For the NMF-$\|\cdot\|_F$ model, we (qualitatively) observe from Table A.5, that topic 12 (“car", “engine", “oil") is related to autos, and topic 4 (“game", “team", “hockey",“baseball") captures other recreation games like hockey and baseball. We observe in Figure A.4 that topics 4 and 12 are associated to the class Recreation. Further, topic 8 (“book",“true",“evidence") relates to atheism, and topic 11 (“god",“jesus",“christ",“faith") relates to religion and specifically Chris-

Table A.5: Top keywords representing each topic of the NMF-$\|\cdot\|_F$ + SVM model referred to in Figure A.4.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Keywords	people gun right government fbi guns law us think batf	key chip clipper encryption keys algorithm escrow phone government number	armenian armenians turkish genocide armenia turks soviet turkey muslim russian	game team games year hockey baseball last players season espn	thanks please mail advance edu e anyone list send address	israel jews israeli arab jewish arabs palestinian palestinians peace israelis	would like think could make say church might someone something	one two book another true point evidence may thing word	x r window server windows motif display running application sun	get mac use space system software drive apple mhz monitor	god jesus christ faith believe sin bible us christians lord	car like cars engine good new v price oil dealer	know anyone need something like us anybody heard sure program
Hard	guns	crypt	mideast	hockey	graphics	mideast	space	atheism	windows	mac	christian	autos	electronics
Score	0.7080	0.5266	0.2576	0.7947	0.2647	0.3854	0.1655	0.2115	0.5781	0.6499	0.5436	0.6138	0.0864
Soft	guns	crypt	mideast	hockey	graphics	mideast	christian	atheism	windows	mac	christian	autos	graphics
Score	0.4408	0.3594	0.2025	0.5635	0.2149	0.2707	0.1426	0.1722	0.4083	0.4519	0.3427	0.4238	0.0929

tianity. Lastly, we observe in Figure A.4 that topic 13 is a shared across 3 classes (Computers, Sciences, and Religion), and is not as informative as the other topics.

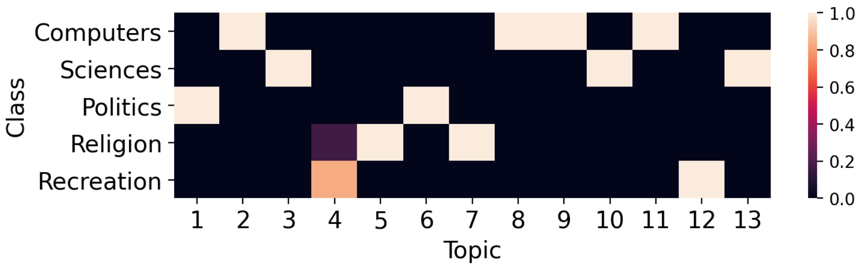

Figure A.5: The coefficients matrix of the SVM classifier (with $D(\cdot\|\cdot)$-NMF) corresponding to the median test classification accuracy equal to 74.89%. Here, all negative coefficients are thresholded to 0, and then each column is normalized to showcases the distribution of the topic over classes

Table A.6: Top keywords representing each topic of the $D(\cdot\|\cdot)$-NMF + SVM model referred to in Figure A.5.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Keywords	armenian armenians turkish p turkey genocide greek armenia turks russian	x r file window edu c windows server display program	sound power one time copy like use problem would signal	get like going time think one back would know really	com deleted think one believe exist say atheism perhaps argument	israel gun people fbi jews israeli guns arab fire would	god jesus church christian bible one christians christ faith religion	list event mailing widget points draw white data graphics call	thanks anyone please know would post mail help someone information	space gas earth would nasa test ground shuttle orbit high	car drive price apple mac k card video buy speed	game team games year baseball hockey players season play win	key chip government clipper phone encryption keys public law security
Hard	mideast	windows	electronics	baseball	atheism	guns	christian	windows	graphics	space	mac	hockey	crypt
Score	0.2860	0.6528	0.3306	0.1884	0.3305	0.4818	0.4819	0.1860	0.2328	0.6014	0.5896	0.7864	0.6910
Soft	mideast	windows	windows	baseball	atheism	guns	christian	windows	graphics	space	mac	hockey	crypt
Score	0.2577	0.5010	0.2424	0.2306	0.2512	0.3281	0.5430	0.1856	0.2301	0.4237	0.4435	0.5902	0.5032

For the $D(\cdot\|\cdot)$-NMF model, we (qualitatively) observe from Table A.6, that topic 3 (“sound",“power") is related to electronics, topic 10 (“space", “gas", “earth") is related to space, topic 13 (“key",“chip",“government") is related to cryptography. All three topics are associated to class Sciences. Topic 12 (“game", “team", “hockey",“baseball") captures recreation games like hockey and baseball and is associated to class Recreation. Further, topic 7 (“god",“jesus",“church") relates to religion and specifically Christianity and is associated to class Religion. Lastly, we observe in Figure A.5 that topic 4 is a shared across Religion and Recreation, and is not as informative as the other topics.

A.3 Clustering Analysis on 20 Newsgroups Dataset

In this section, we measure the performance of the NMF and SSNMF topic models with a clustering-motivated score. In these experiments, we measure the similarity of ground-truth clusters, encoded by a given label matrix $\mathbf{M}$, to NMF/SSNMF computed clusters, encoded by the representation matrix $\mathbf{S}$. We denote by $\mathbf{M}$ the (column-wise) one-hot encoded label matrix which maps documents to the subgroups to which they belong[13], $\mathbf{M} \in \{0,1\}^{13\times 8980}$ (subgroups by documents). We let $\mathbf{S}$ be the representation matrix computed by NMF/SSNMF, in which the ith row provides the association of each document with the ith topic.

We employ two approaches to clustering or mixture assignment. The first is *hard clustering* in which the documents are assigned to a single cluster corresponding to computed topics. In this approach, we apply a mask to the representation matrix, $\hat{\mathbf{S}} = \text{label}(\mathbf{S})$, where $\text{label}(\cdot)$ assigns the largest entry of each column to 1 and all other entries to 0. The second approach is *soft clustering* in which the documents are assigned to a distribution of clusters corresponding to the topics. In this approach, we normalize each of the columns of the representation matrix to have sum 1 to produce $\hat{\mathbf{S}}$.

Now, in either approach, we apply a metric P which measures the association between the ℓth topic-documents association $\hat{\mathbf{S}}_{\ell,\bullet}$ and the best ground truth subgroup-documents association, $\mathbf{M}_{I,\bullet}$ that is, for topic ℓ, we define I as

[13] In the 20 Newsgroups dataset, each document belongs to only one subgroup.

$$I = \operatorname*{argmax}_{\mathbf{i}} \frac{\|\hat{\mathbf{S}}_{\ell,\bullet} \odot \mathbf{M}_{i,\bullet}\|_1}{\|\mathbf{M}_{i,\bullet}\|_1},$$

and define score P for the ℓth topic as

$$P(\hat{\mathbf{S}}_{\ell,\bullet}) = \frac{\|\hat{\mathbf{S}}_{\ell,\bullet} \odot \mathbf{M}_{I,\bullet}\|_1}{\|\mathbf{M}_{I,\bullet}\|_1},$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm. We note that this metric is similar to that of [168]; we use score P instead as it allows us to measure clustering performance topic-wise. We also note that the learned topics of NMF and SSNMF methods need not be in one-to-one correspondence with the subgroups in Table 3.2 as topics are also learnt for the classification task at hand.

We present in Table A.7 the average (averaged over topics) score P for the representation matrices computed by each of the NMF/SSNMF models in both the hard-clustering and soft-clustering settings. The scores P for each topic (for both hard-clustering and soft-clustering) and the maximizing subgroup (indicated by I) are listed in the bottom four rows of the keyword table associated to each model; see the last four rows of Tables A.2-A.6, and A.8.

Table A.7: Listed scores are average over 11 trials; in each trial, we average score P across all topics.

Model	Hard Clustering	Soft Clustering
$(\Vert\cdot\Vert_F, \Vert\cdot\Vert_F)$	0.3895	0.3647
$(\Vert\cdot\Vert_F, D(\cdot\Vert\cdot))$	0.3857	0.3703
$(D(\cdot\Vert\cdot), \Vert\cdot\Vert_F)$	0.3874	0.3553
$(D(\cdot\Vert\cdot), D(\cdot\Vert\cdot))$	0.3874	0.3732
$\Vert\cdot\Vert_F$-NMF	0.4348	0.3080
$D(\cdot\Vert\cdot)$-NMF	**0.4879**	**0.3764**

Table A.8: Clustering results for topics of the $(D(\cdot\Vert\cdot), \Vert\cdot\Vert_F)$-SSNMF model referred to in Figure 3.2.

Topics	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
Hard	crypt	autos	christian	windows	guns	hockey	hockey	mideast	electronics	space	mideast	space	christian
Score	0.5698	0.4167	0.7332	0.9867	0.5693	0.3046	0.3146	0.1976	0.3106	0.0692	0.2140	0.1115	0.2477
Soft	crypt	autos	christian	windows	mideast	hockey	baseball	mideast	electronics	crypt	mideast	electronics	christian
Score	0.3284	0.3260	0.5527	0.9181	0.4010	0.3006	0.3056	0.2689	0.2733	0.1577	0.2601	0.2217	0.3571