



Article

Continuous Semi-Supervised Nonnegative Matrix Factorization

Michael R. Lindstrom ^{1,*}, Xiaofu Ding ², Feng Liu ², Anand Somayajula ²  and Deanna Needell ² 

¹ School of Mathematical and Statistical Sciences, The University of Texas Rio Grande Valley, Edinburg, TX 78539, USA

² Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095, USA

* Correspondence: mike.lindstrom@utrgv.edu

Abstract: Nonnegative matrix factorization can be used to automatically detect topics within a corpus in an unsupervised fashion. The technique amounts to an approximation of a nonnegative matrix as the product of two nonnegative matrices of lower rank. In certain applications it is desirable to extract topics and use them to predict quantitative outcomes. In this paper, we show Nonnegative Matrix Factorization can be combined with regression on a continuous response variable by minimizing a penalty function that adds a weighted regression error to a matrix factorization error. We show theoretically that as the weighting increases, the regression error in training decreases weakly. We test our method on synthetic data and real data coming from Rate My Professors reviews to predict an instructor's rating from the text in their reviews. In practice, when used as a dimensionality reduction method (when the number of topics chosen in the model is fewer than the true number of topics), the method performs better than doing regression after topics are identified—both during training and testing—and it retrains interpretability.

Keywords: topic modelling; regression; nonnegative matrix factorization; optimization



Citation: Lindstrom, M.R.; Ding, X.; Liu, F.; Somayajula, A.; Needell, D. Continuous Semi-Supervised Nonnegative Matrix Factorization. *Algorithms* **2023**, *16*, 187. <https://doi.org/10.3390/a16040187>

Academic Editors: Aneasha Bakharia, Khanh Luong and Frank Werner

Received: 15 January 2023

Revised: 21 February 2023

Accepted: 28 February 2023

Published: 30 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nonnegative matrix factorization (NMF) is a highly versatile data science technique with far-reaching applications. It can identify thematic elements, i.e., groups of words that appear frequently together in a corpus, which together convey a common message [1]. More generally, it can be used to decompose an image into identifiable patterns [2] and as a general-purpose dimensionality reduction or preprocessing method before applying other machine learning methods, as has been done in studying various diseases [3,4]. Similar to singular value decomposition (SVD) [5], NMF provides a low rank factorization. In NMF, a nonnegative matrix $X \in \mathbb{R}_{\geq 0}^{n \times m}$ representing a corpus (or other nonnegative dataset) is factored into a low rank approximation $X \approx WH$ where the inner dimension, r , between W and H is such that $r \ll m$ and $r \ll n$; however, unlike SVD, there is an additional constraint that both W and H are nonnegative, i.e., $W \in \mathbb{R}_{\geq 0}^{n \times r}$ and $H \in \mathbb{R}_{\geq 0}^{r \times m}$. This nonnegativity enforces that the data in X are represented by a nonnegative combination of the dictionary atoms in the factorization, which lends itself to human interpretability. For example, in the foundational work [2], Lee and Seung show that NMF, when applied to facial images, decomposes the images into recognizable parts such as noses, eyes, and mouths.

When applied to a document-term matrix [6] X , where row i of X represents document i and column j represents the frequencies of word j across the documents, the classical NMF method amounts to

$$(W, H) = \arg \min_{W \in \mathbb{R}_{\geq 0}^{n \times r}, H \in \mathbb{R}_{\geq 0}^{r \times m}} \|X - WH\|_F^2 \quad (1)$$

where, for $A \in \mathbb{R}^{n \times m}$, $\|A\|_F$ denotes the Frobenius norm of A with $\|A\|_F^2 = \text{tr}(A^T A) = \sum_{i=1}^n \sum_{j=1}^m |A_{ij}|^2$. Other variations on the penalty function exist, including the Kullback–Leibler divergence [7]. After computing W and H , we interpret row j of H as the j th

topic, its components being the weight of each word in that topic, and row i of W as the topic-encoding of document i , i.e.,

$$X_{i,:} \approx \sum_{j=1}^r W_{i,j} H_{j,:}.$$

Throughout this manuscript we make use of “colon notation” where “:” means the full range of indices for a row/column, “a:b” indicates a consecutive range of indices from a to b , etc.

The rest of our paper is organized as follows: the general context to our work and our main contributions are stated in Section 2; in Section 3, we provide the formulation of our method, its algorithmic implementation, its theoretical properties, and their proofs; in Section 4, we provide a proof of concept through synthetic data; in Section 5, we test our method on a real dataset from Rate My Professors; and finally we conclude our work in Section 6.

2. Relation to Current Work and Contributions

When designing algorithms to handle multiple objectives simultaneously, a weighted penalty function that combines the multiple objectives is often used [8]. For example, with LASSO regression [9], one seeks a linear model that also does model selection by zeroing out parameters that are less significant. This can be done by minimizing a least squares error with a weighted penalty for the ℓ^1 -norm of the parameters. Prior authors have combined NMF with a linear regression procedure to maximize the predictive power of a classifier [10–12]. This is accomplished through a penalty function that combines NMF with another objective function—a (semi) supervised approach. Semi-supervised NMF can also be applied to guide NMF to identify topics with desired keywords [13].

In this paper, we combine NMF with a linear regression model to predict the value of a continuous response variable. We consider synthetic data along with a real dataset that pairs written commentary with a real-valued observation. In particular, we use reviews from the Rate My Professors website [14,15] that include all student comments for a professor along with the mean rating in [1, 5]. Due to the averaging, the rating is effectively a continuous variable.

3. Model

We provide the framework for our proposed continuous semi-supervised nonnegative matrix factorization method (CSSNMF).

3.1. Formulation

We consider having a document-term matrix [6] $X \in \mathbb{R}_{\geq 0}^{n \times m}$ for n documents with their associated word frequencies in the m columns—a “bag of words” where each document is represented only by the frequencies of its words. Each document has a corresponding value in \mathbb{R} so that we can associate with X the vector $Y \in \mathbb{R}^{n \times 1}$. Put another way: each document is represented as a row of X , call it $x \in \mathbb{R}_{\geq 0}^{1 \times m}$, which stores the frequencies of each of the m words within the corpus; then, to each such x there is an observation $y \in \mathbb{R}$ (and over n documents, this generates $Y \in \mathbb{R}^{n \times 1}$). We choose $r \in \mathbb{N}$ and $\lambda \geq 0$ as hyper-parameters where r denotes the number of topics and λ is weight put on the regression error.

In the real data that we look at, each row of X will represent the reviews written for one university instructor, with frequencies of the words in the columns. For each instructor in the dataset, the mean value of their respective student ratings will be a single component of Y . From a predictive standpoint, we would like to *predict the mean rating* of an instructor *based only on the words* in their review, i.e., take a vector $x \in \mathbb{R}_{\geq 0}^{1 \times m}$ of the word frequencies and make a prediction \hat{y} of their mean rating (the hat indicates a prediction). We want the prediction \hat{y} to be as close to the true mean rating as possible. The topic modelling aspect of this is that instead of using the full x vector of dimension m , we approximate

x as a nonnegative linear combination of r topic vectors (interpretable vectors of word frequencies). We effectively compress x to a vector $w \in \mathbb{R}_{\geq 0}^{1 \times r}$ of dimension r , and we model the rating as a linear combination of the components of w .

Given $W \in \mathbb{R}_{\geq 0}^{n \times r}$, $H \in \mathbb{R}_{\geq 0}^{r \times m}$, and $\theta \in \mathbb{R}^{(r+1) \times 1}$, we define a penalty function that combines topic modelling with a linear regression based on the topic representations. The intuition with the weighting is that as the weight λ increases, topic modelling is still done, but more and more emphasis is put on producing an accurate regression on Y . We define

$$F^{(\lambda)}(W, H, \theta; X, Y) = N(W, H; X) + \lambda R(W, \theta; Y), \text{ where} \quad (2)$$

$$N(W, H; X) := \|X - WH\|_F^2 \quad (3)$$

$$R(W, \theta; Y) := \|\tilde{W}\theta - Y\|^2 \quad (4)$$

and where $\tilde{W} \in \mathbb{R}^{n \times (r+1)}$ is given by

$$\tilde{W} := \begin{pmatrix} 1 & W_{1,1} & \dots & W_{1,m} \\ 1 & W_{2,1} & \dots & W_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_{n,1} & \dots & W_{n,m} \end{pmatrix}. \quad (5)$$

The matrix \tilde{W} with its column of 1s allows for an intercept: given a topic representation $w \in \mathbb{R}^{1 \times r}$, we predict a value $\hat{y} = \theta_1 + \theta_2 w_1 + \dots + \theta_{r+1} w_r$.

When $\lambda > 0$, we seek

$$(W^{(\lambda)}, H^{(\lambda)}, \theta^{(\lambda)}) = \arg \min_{W, H, \theta} F^{(\lambda)}(W, H, \theta; X, Y). \quad (6)$$

When $\lambda = 0$, we define

$$(W^{(0)}, H^{(0)}) = \arg \min_{W, H} N(W, H; X) \quad (7)$$

$$\theta^{(0)} = \arg \min_{\theta} R(W^{(0)}, \theta; Y). \quad (8)$$

We also impose a normalization constraint, that

$$\forall i, \quad \sum_{j=1}^m H_{ij} = 1 \quad (9)$$

so that the topics have unit length in ℓ^1 .

Remark 1 (Sum of Topic Representations). If $X \approx WH$ is normalized so its rows sum to 1 then it is also the case that $\forall i, \quad \sum_{j=1}^r W_{ij} \approx 1$ by noting that $X_{ij} = \sum_{k=1}^r W_{ik} H_{kj}$ and summing over j .

When $\lambda = 0$, θ has no effect upon $F^{(\lambda)}$ and we first perform regular NMF over W and H and, as a final step, we choose θ to minimize the regression error. In other words, if $\lambda = 0$, we do NMF first and then find the best θ given the already determined weights for each document. It seems intuitive, however, that the regression could be improved if θ and W both were being influenced by the regression to Y , which is what our method aims to do when $\lambda > 0$. From a practical perspective, if $\lambda \uparrow \infty$, then the regression error becomes dominant and we may expect the topics as found in H to be less meaningful. In Section 3.2, we state some theoretical properties of our method as it is being trained.

Once $H^{(\lambda)}$ and $\theta^{(\lambda)}$ are known, we can make predictions for the response variable corresponding to a document. This amounts to finding the best nonnegative topic encoding $w \in \mathbb{R}_{\geq 0}^{1 \times r}$ for the document and using that encoding in the linear model—see Section 3.3.

Remark 2 (Uniqueness). Using our established notation, we remark that if $X^* = WH$ and $Y^* = \theta_1 + W\theta_{2:(r+1)}$, then $X^* = \tilde{W}\tilde{H}$ and $Y^* = \theta_1 + \tilde{W}\tilde{\theta}$, where $\tilde{W} = WS$, $\tilde{H} = S^{-1}H$, and

$\tilde{\theta} = S^{-1}\theta_{2:(r+1)}$ for any invertible $S \in \mathbb{R}^{r \times r}$ with SW and $S^{-1}H$ both having all entries nonnegative. Thus, uniqueness of an optima, if it exists, can only be unique up to matrix multiplications.

3.2. Theoretical Results

We present two important behaviors of CSSNMF with regard to increasing λ and its effect upon predicting the response variable.

Proposition 1 (Regression Error with Nonzero λ). For $\lambda \geq 0$, let $W^{(\lambda)}, H^{(\lambda)}, \theta^{(\lambda)}$ be a unique (as per Remark 2) global minimum to Equations (6)–(9). Then $R(W^{(\lambda)}, \theta^{(\lambda)}) \leq R(W^{(0)}, \theta^{(0)})$.

Theorem 1 (Weakly Decreasing Regression Error). Let $0 \leq \lambda_1 < \lambda_2$ be given where $W^{(\lambda_i)}, H^{(\lambda_i)}, \theta^{(\lambda_i)}$ are the unique (as per Remark 2) global minimizers of Equations (6)–(9) for $i = 1, 2$. Then $R(W^{(\lambda_2)}, \theta^{(\lambda_2)}; Y) \leq R(W^{(\lambda_1)}, \theta^{(\lambda_1)}; Y)$.

Remark 3. Proposition 1 and Theorem 1 are based on obtaining a global minimum. In practice, we may only find a local minimum.

Proposition 1 and Theorem 1 are statements pertaining to training the model. Assuming we have the optimal solutions, Proposition 1 tells us that the regression error for $\lambda > 0$ is no worse than the regression error with $\lambda = 0$ and could, in fact, be better. Thus, the intuition that selecting topics while paying attention to the regression error is practical. Then Theorem 1 says that the regression error is weakly monotonically decreasing as λ increases. In practical application, we find the error strictly monotonically decreases.

Before proceeding to algorithmic procedures, we prove Proposition 1 and Theorem 1.

Proof of Proposition 1. If $\lambda > 0$ then

$$\begin{aligned} F^{(\lambda)}(W^{(\lambda)}, H^{(\lambda)}, \theta^{(\lambda)}; X, Y) &\leq F^{(\lambda)}(W^{(0)}, H^{(0)}, \theta^{(0)}; X, Y) \implies \\ N(W^{(\lambda)}, H^{(\lambda)}; X) + \lambda R(W^{(\lambda)}, \theta^{(\lambda)}; Y) &\leq N(W^{(0)}, H^{(0)}; X) + \lambda R(W^{(0)}, \theta^{(0)}; Y) \\ \implies \\ \lambda(R(W^{(\lambda)}, \theta^{(\lambda)}; Y) - R(W^{(0)}, \theta^{(0)}; Y)) &\leq N(W^{(0)}, H^{(0)}; X) - N(W^{(\lambda)}, H^{(\lambda)}; X) \\ &\leq 0. \end{aligned}$$

The first inequality comes from how $(W^{(\lambda)}, H^{(\lambda)}, \theta^{(\lambda)})$ are defined by Equation (6). The final inequality comes from how $(W^{(0)}, H^{(0)})$ are defined as minimizers in Equation (7).

Since we first assumed $\lambda > 0$, we obtain $R(W^{(\lambda)}, \theta^{(\lambda)}; Y) \leq R(W^{(0)}, \theta^{(0)}; Y)$. Finally, if $\lambda = 0$, then there is equality with $R(W^{(\lambda)}, \theta^{(\lambda)}; Y) = R(W^{(0)}, \theta^{(0)}; Y)$. \square

Proof of Theorem 1. Note that if $\lambda_1 = 0$, then Theorem 1 already applies, so we assume $0 < \lambda_1 < \lambda_2$. We have that

$$\begin{aligned} F^{(\lambda_1)}(W^{(\lambda_1)}, H^{(\lambda_1)}, \theta^{(\lambda_1)}; X, Y) &\leq F^{(\lambda_1)}(W^{(\lambda_2)}, H^{(\lambda_2)}, \theta^{(\lambda_2)}; X, Y) \implies \\ \lambda_1(R(W^{(\lambda_1)}, \theta^{(\lambda_1)}; Y) - R(W^{(\lambda_2)}, \theta^{(\lambda_2)}; Y)) &\leq \\ N(W^{(\lambda_2)}, H^{(\lambda_2)}; X) - N(W^{(\lambda_1)}, H^{(\lambda_1)}; X). \end{aligned} \quad (10)$$

We also have

$$\begin{aligned} \lambda_2(R(W^{(\lambda_2)}, \theta^{(\lambda_2)}; Y) - R(W^{(\lambda_1)}, \theta^{(\lambda_1)}; Y)) &\leq \\ N(W^{(\lambda_1)}, H^{(\lambda_1)}; X) - N(W^{(\lambda_2)}, H^{(\lambda_2)}; X). \end{aligned} \quad (11)$$

Adding Equations (10) and (11) together,

$$(\lambda_1 - \lambda_2)R(W^{(\lambda_1)}, \theta^{(\lambda_1)}; Y) + (\lambda_2 - \lambda_1)R(W^{(\lambda_2)}, \theta^{(\lambda_2)}; Y) \leq 0$$

which, upon dividing by $\lambda_2 - \lambda_1 > 0$, directly gives

$$R(W^{(\lambda_2)}, \theta^{(\lambda_2)}; Y) \leq R(W^{(\lambda_1)}, \theta^{(\lambda_1)}; Y).$$

□

3.3. Algorithm

Our minimization approach is iterative and based on the alternating nonnegative least squares [16] approach. Due to the coupling of NMF and regression errors, other approaches such as multiplicative or additive updates [17] are less natural. Each iteration consists of: (1) holding H and θ fixed while optimizing each row of W separately (nonnegative least squares); (2) holding W and θ fixed while optimizing each column of H separately (nonnegative least squares); and finally (3) holding W and H fixed while optimizing over θ . The error is nondecreasing between iterations and from one optimization to the next. We now derive and justify this approach (Algorithm 1) in increasing complexity of cases.

Algorithm 1: Overall CSSNMF algorithm.

Input : A matrix $X \in \mathbb{R}_{\geq 0}^{n \times m}$,

a vector $Y \in \mathbb{R}^{n \times 1}$,

a positive integer $r \in \mathbb{N}$,

a scalar $\lambda \geq 0$,

a relative error tolerance $\tau > 0$, and

a maximum number of iterations $maxIter$.

Output: Minimizers of Equations (6)–(9): nonnegative matrix $W \in \mathbb{R}_{\geq 0}^{n \times r}$,

nonnegative matrix $H \in \mathbb{R}_{\geq 0}^{r \times m}$, and

vector $\theta \in \mathbb{R}^{(r+1) \times 1}$.

1 $relErr = \infty, err = \infty$

2 Elementwise, $W \sim Unif([0, \|X\|_\infty))$, $H \sim Unif([0, \|X\|_\infty))$,
 $\theta \sim Unif([0, \|X\|_\infty))$.

3 $iter = 0$

4 **while** $relErr > \tau$ **and** $iter < maxIter$ **do**

5 $W \leftarrow newW$ as per Algorithm 2

6 $H \leftarrow newH$ as per Algorithm 3

7 $\theta \leftarrow new\theta$ as per Algorithm 4

8 Normalize W , H , and θ as per Algorithm 5

9 $errTemp = F^{(\lambda)}(W, H, \theta; X, Y)$

10 **if** $err < \infty$ **then**

11 $relErr \leftarrow |err - errTemp| / err$

12 **end if**

13 $err \leftarrow errTemp$

14 $iter \leftarrow iter + 1$

15 **end while**

16 **return** W, H, θ

Algorithm 2: Updating W .

Input : A matrix $X \in \mathbb{R}_{\geq 0}^{n \times m}$,
 a vector $Y \in \mathbb{R}^{n \times 1}$,
 a matrix $W \in \mathbb{R}_{\geq 0}^{n \times r}$,
 a matrix $H \in \mathbb{R}_{\geq 0}^{r \times m}$,
 and a scalar $\lambda \geq 0$.
Output: A new value for W .
 1 $\bar{\theta} = (\theta_2, \dots, \theta_{r+1})^T$
 2 $\bar{X} = \begin{bmatrix} X & | & \sqrt{\lambda}(\theta_1 - Y) \end{bmatrix}$
 3 $\bar{W} = \begin{bmatrix} H & | & \sqrt{\lambda}\bar{\theta} \end{bmatrix}$
 4 **for** $i \leftarrow 1 \dots n$ **do**
 5 $W_{i,:} \leftarrow \arg \min_{w \in \mathbb{R}_{\geq 0}^{1 \times r}} \|\bar{X}_{i,:} - w\bar{H}\|^2$
 6 **end for**
 7 **return** W

Algorithm 3: Updating H .

Input : A matrix $X \in \mathbb{R}_{\geq 0}^{n \times m}$,
 a matrix $W \in \mathbb{R}_{\geq 0}^{n \times r}$, and
 a matrix $H \in \mathbb{R}_{\geq 0}^{r \times m}$.
Output: A new value for H .
 1 **for** $j \leftarrow 1 \dots m$ **do**
 2 $H_{:,j} \leftarrow \arg \min_{h \in \mathbb{R}_{\geq 0}^{r \times 1}} \|X_{:,j} - Wh\|^2$
 3 **end for**
 4 **return** H

Algorithm 4: Updating θ .

Input : A vector $Y \in \mathbb{R}^{n \times 1}$, and
 a matrix $W \in \mathbb{R}_{\geq 0}^{n \times r}$.
Output: A new value for θ .
 1 $e = (1, 1, \dots, 1)^T \in \mathbb{R}^{n \times 1}$
 2 $\bar{W} = \begin{bmatrix} e & | & W \end{bmatrix}$
 3 **return** $\bar{W}^+ Y$

Algorithm 5: Normalization process.

Input : A matrix $W \in \mathbb{R}_{\geq 0}^{n \times r}$,
 a matrix $H \in \mathbb{R}_{\geq 0}^{r \times m}$, and
 a vector $\theta \in \mathbb{R}^{(r+1) \times 1}$.
Output: New values for W , H , and θ .
 1 $S \in \mathbb{R}_{\geq 0}^{r \times 1}$ a vector of row sums of H .
 2 $S \leftarrow \text{diag}(S)$
 3 $W \leftarrow WS$.
 4 $H \leftarrow S^{-1}H$.
 5 $\theta_{2:(r+1)} \leftarrow S^{-1}\theta_{2:(r+1)}$.
 6 **return** W , H , and θ .

W and H fixed.

If W and H are given and only θ can vary, then Equation (2) is minimized when $\|\bar{W}\theta - Y\|^2$ is minimized. If \bar{W} has full rank or is overdetermined, this happens when the error, $\bar{W}\theta - Y$, is orthogonal to the column span of \bar{W} or that

$$\theta = (\bar{W}^T \bar{W})^{-1} \bar{W}^T Y. \quad (12)$$

When \bar{W} is underdetermined or has full rank, we require $\bar{W}\theta = Y$ with $\|\theta\|$ minimized (for uniqueness) whereby

$$\theta = \bar{W}^T (\bar{W} \bar{W}^T)^{-1} Y. \quad (13)$$

See Algorithm 4. Thus, when \bar{W} does not have full rank, the solution is $\theta = \bar{W}^+ Y$ where \bar{W}^+ is the pseudoinverse of \bar{W} . See [18] for a discussion of pseudoinverses. In applications, we only expect to see overdetermined systems because the number of topics r should be less than the number of documents n .

W and θ fixed.

If W and θ are given and only H can change, then minimizing Equation (2) requires minimizing $N(W, H; X)$. We can expand this error term out in the columns of H :

$$\begin{aligned} N(W, H; X) &= \|X - WH\|_F^2 \\ &= \sum_{j=1}^m \|(X - WH)_{:,j}\|^2 \\ &= \sum_{j=1}^m \|X_{:,j} - WH_{:,j}\|^2. \end{aligned}$$

Because columnwise the terms of the sum are independent, we can minimize each column $H_{:,j}$ of H separately to minimize the sum, i.e.,

$$H_{:,j} = \arg \min_{h \in \mathbb{R}_{\geq 0}^{r \times 1}} \|X_{:,j} - Wh\|^2, \quad j = 1, 2, \dots, m, \quad (14)$$

as given in Algorithm 3.

H and θ fixed.

When H and θ are fixed, then Equation (2) can be written out as

$$\begin{aligned} F^{(\lambda)} &= \|X - WH\|_F^2 + \lambda \|\bar{W}\theta - Y\|^2 \\ &= \sum_{i=1}^n \|(X - WH)_{i,:}\|^2 + \lambda \sum_{i=1}^n (\bar{W}\theta - Y)_i^2 \\ &= \sum_{i=1}^n \|X_{i,:} - W_{i,:}H\|^2 + \sum_{i=1}^n \left(\sqrt{\lambda} (\theta_1 e + W_{i,:}\bar{\theta} - Y) \right)_i^2 \end{aligned} \quad (15)$$

where $e = (1, 1, \dots, 1)^T \in \mathbb{R}^{n \times 1}$ and $\bar{\theta} = (\theta_2, \dots, \theta_{r+1})^T \in \mathbb{R}^{r \times 1}$. Defining matrices

$$\tilde{X} = \begin{bmatrix} X & | & \sqrt{\lambda}(\theta_1 e - Y) \end{bmatrix} \quad (16)$$

$$\tilde{H} = \begin{bmatrix} H & | & \sqrt{\lambda}\bar{\theta} \end{bmatrix} \quad (17)$$

we can rewrite Equation (15) as

$$F^{(\lambda)} = \sum_{i=1}^n \|\tilde{X}_{i,:} - W_{i,:}\tilde{H}\|^2,$$

which can be minimized through

$$W_{i,:} = \arg \min_{w \in \mathbb{R}_{\geq 0}^{1 \times r}} \|\bar{X}_{i,:} - w\bar{H}\|^2, \quad i = 1, 2, \dots, n. \quad (18)$$

This is precisely Algorithm 2.

For our optimizations and linear algebra, we used Numpy [19] and SciPy [20]. The nonnegative least squares routine employs an active set method to solve the least squares minimization problem with inequality constraints [21]. The active set method amounts to gradually building up the set of active constraints (those for which their not being enforced in the unconstrained problem would result in constraint violation or equality, i.e., a regression variable with a nonnegativity constraint being less than or equal to 0) and then optimizing over the passive set (all variables not in the active set) once identified with equality constraints on the active set [22]. This method can also be parallelized [23].

In addition to the steps outlined within these algorithms, we employed two additional modifications: (1) we defined $\epsilon = 10^{-10}$, and any entries in H less than ϵ were replaced by ϵ (otherwise on some occasions, the W update step would fail); and (2) the minimizations at times yielded worse objective errors than already obtained and, when this happened, we did not update to the worse value.

As noted with other NMF routines, we might not reach a global minimizer [24]. In practice, the minimization should be run repeatedly with different random initializations to find a more ideal local minimum.

From an application standpoint, we wish to run the model on documents it has not been trained on. Algorithm 6 stipulates how a prediction takes place. We first find the best nonnegative decomposition of the document, a vector in $\mathbb{R}^{1 \times m}$, into the topic basis, projecting to r -dimensions. With the representation in topic-coordinates, we then use the linear model.

Algorithm 6: Prediction process.

Input : A matrix $H \in \mathbb{R}_{\geq 0}^{r \times m}$,
a vector $\theta \in \mathbb{R}^{(r+1) \times 1}$,
and a vector $x \in \mathbb{R}^{1 \times m}$.

Output: Model prediction for response variable, \hat{y} .

- 1 Compute $w = \arg \min_{w \in \mathbb{R}_{\geq 0}^{1 \times r}} \|wH - x\|^2$.
 - 2 Compute $\hat{y} = \theta_1 + w\theta_{2:(r+1)}$.
 - 3 **return** \hat{y}
-

An implementation of our algorithm can be found on our [BitBucket repository](#), accessed on 21 February 2023.

4. Synthetic Datasets

In our synthetic data, we generate a matrix X that has nonnegative factors W and H , but we add noise. We also generate a response vector Y given as the matrix–vector product $\bar{W}\theta$ with noise. We investigate four items: (1) that the method does in fact work to decrease the objective function; (2) whether the regression errors decrease with increasing λ ; (3) the effects of overfitting; and (4) the model robustness to noise.

4.1. Generating Synthetic Data

Our synthetic data generation can be summarized as follows:

1. We fix values of $n = 100$, $m = 40$, $M = 20$, and $r = 4$.
2. We then define $\eta_x = \eta_y = 4$.
3. We pick $X \in \mathbb{R}^{n \times r}$ such that each entry is $\sim \text{Unif}([0, M])$. We likewise choose $H \in \mathbb{R}^{r \times m}$.

4. We set $X = WH$.
5. We pick $\theta \in \mathbb{R}^{(r+1) \times 1}$ such that each element is $\sim \text{Unif}([-M/2, M/2])$.
6. We set $Y = \bar{W}\theta$.
7. We perturb X with noise $\sim \mathcal{D}_X$ and Y with noise $\sim \mathcal{D}_Y$.
8. Any negative X -entries are set to 0.

We consider two different forms for \mathcal{D}_X and \mathcal{D}_Y :

- Being elementwise $\sim \mathcal{N}(0, \eta_x^2)$ and $\sim \mathcal{N}(0, \eta_y^2)$ or
- Being elementwise $\sim \text{Unif}([0, \eta_x])$ and $\sim \text{Unif}([0, \eta_y])$.

Note that in the synthetic data, the true number of topics is $r = 4$. In testing our synthetic data, we run Algorithm 1 where $\tau = 10^{-4}$ and $\text{maxIter} = 100$. We use 70% of the data for training and 30% for testing.

4.2. Investigation

We confirm that the error in the objective function $F^{(\lambda)}$ decreases with each iteration of Algorithm 1 in Figure 1—done with Gaussian noise.

With the regression error being the mean squared prediction error, from Figure 2, we see the regression error in the training does tend to decrease with λ . (There are a few small exceptions, which we believe stem from randomizations leading to an assortment of different local optima.) The overall scale of the testing errors gets smaller as r goes from 1 to 4 and then stays steady or even gets slightly worse as r increases from 4. Indeed, $r = 4$ is the “correct” synthetic value. Given the noise as either Gaussian or uniform, the variances of $\mathcal{N}(0, \eta_y^2)$, η_y^2 , and $\text{Unif}([0, \eta_y])$, $\eta_y^2/12$, serve as loose estimates for the best possible testing loss (the loss could very well be higher as noise is added to the matrix X as well). When the training errors are smaller than this estimate, it suggests overfitting.

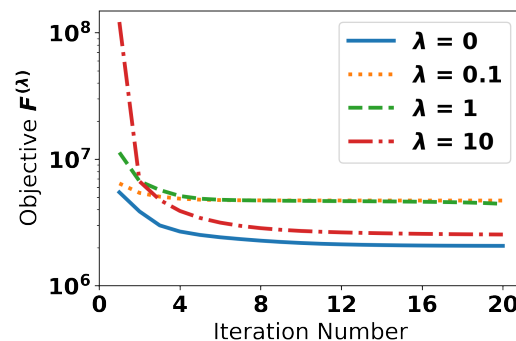


Figure 1. Illustration of decreasing objective function at $r = 3$ topics for $\lambda \in \{0\} \cup \{10^{i/2} | i \in \mathbb{Z} \cap [-2, 2]\}$.

To study robustness with noise, we allow the level Gaussian noise to vary in the problem and evaluate the regression error on testing data. To ensure each noise level starts with the same ground truth, we start with unperturbed X and Y (as per Section 4.1 with $\eta_x = \eta_y = 0$) and compute matrices and vectors with the same size as X and Y with elementwise $\mathcal{N}(0, 1)$ entries. Then, for each level of noise under investigation, we scale these unit normal distributions by a noise level $\eta \in \{0, 4, 8, 12, 16, 20\}$ and examine the testing error as the λ varies with $r = 3$ (rank below true rank), $r = 4$ (correct rank), and $r = 5$ (number of chosen topics above correct rank). Figure 3 illustrates the results. The noise is handled well with $r = 3$ in that a higher λ does tend to improve the testing error; however, at higher noise levels η , it seems suitable minima are harder to find when λ is large. With $r = 4$ and $r = 5$, the testing does not benefit with increasing λ at any noise level.

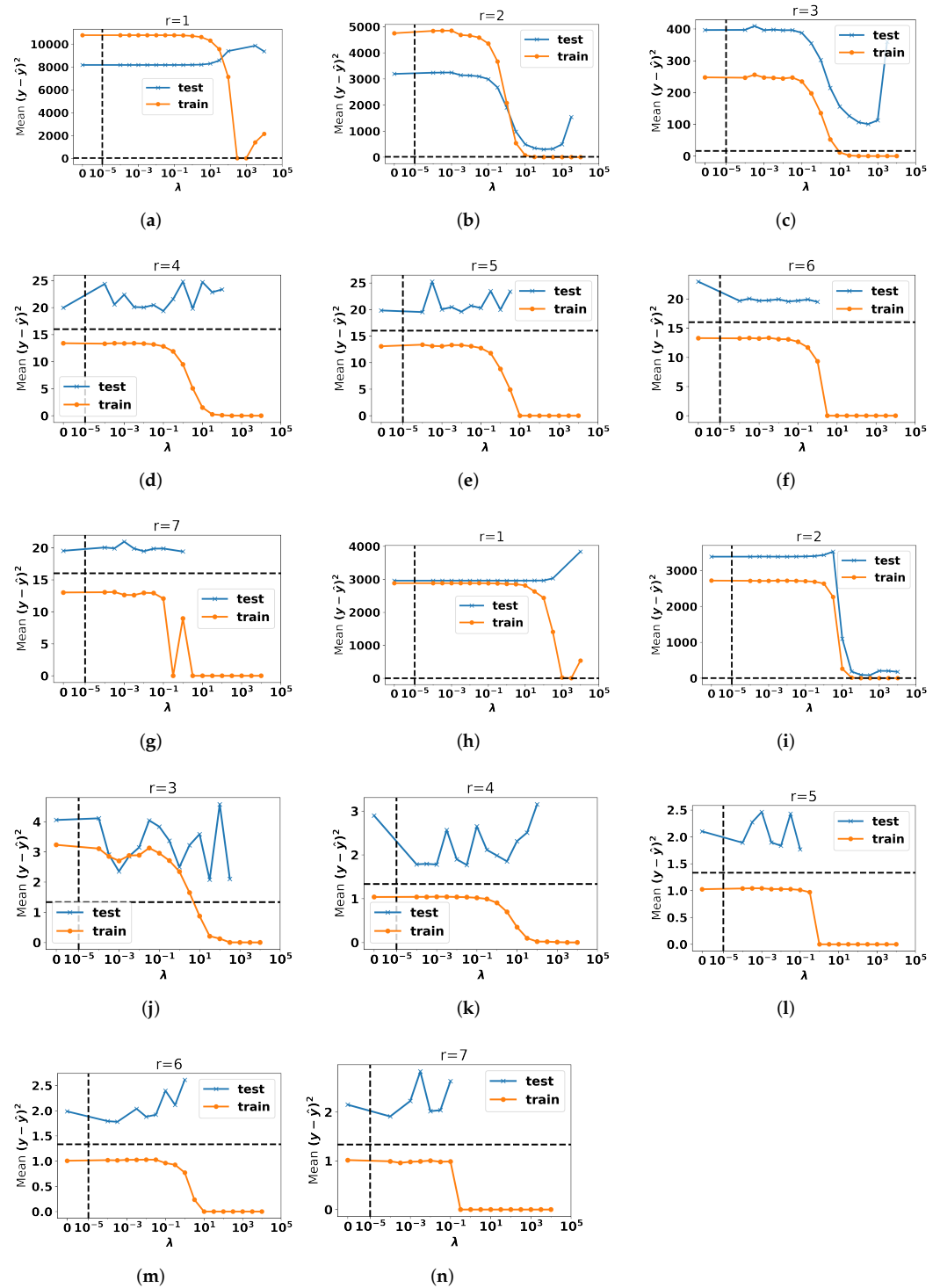


Figure 2. Regression errors with varying regression weight λ with different numbers of topics r for Gaussian noise (a–g) and uniform noise (h–n). The λ values used are the set $\{0\} \cup \{10^{i/2} | i \in \mathbb{Z} \cap [-8, 8]\}$. For each λ and r , fifty trials were run and the regression errors corresponding to the best overall objective function $F^{(\lambda)}$ were recorded. Points with $\lambda > 0$ for which the regression error exceeds 1.5 times the regression error at $\lambda = 0$ are not displayed. The dashed horizontal line is the estimated minimal mean regression error. The dashed vertical line is the transition point between a linear and logarithmic x -scale.

Taken together, we anticipate that CSSNMF will perform well provided the number of topics chosen does not exceed the true number of topics in the dataset (difficult to assess). We expect that the optimal predictions on unseen data should occur at a λ large enough that the testing errors have decreased and plateaued. In Figure 2, we see that for large λ , when overfitting is an issue, the testing performance is seldom better than where $\lambda = 0$ (classical NMF and then regression) and, in fact, is often much worse.

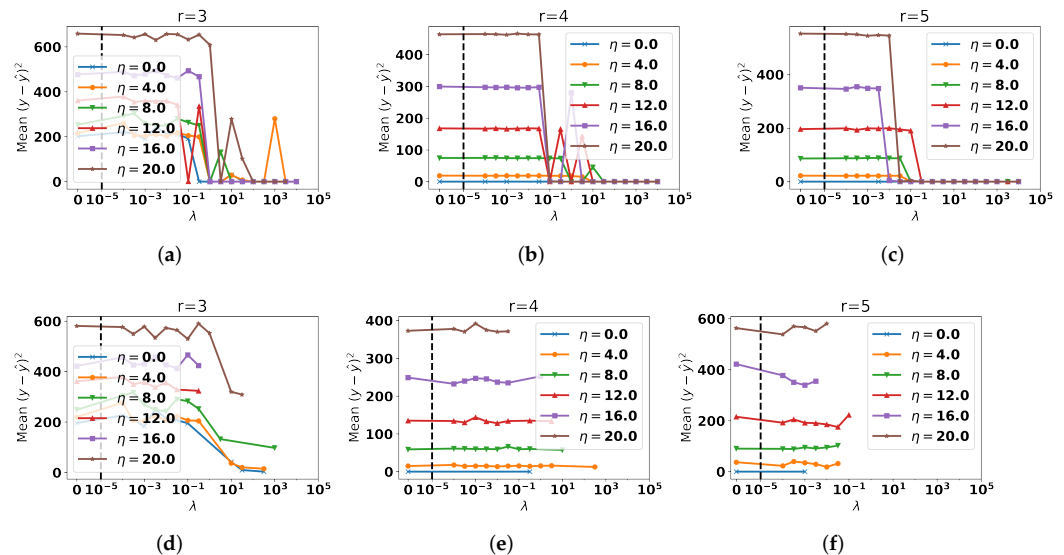


Figure 3. Regression errors at various noise levels η with varying regression weight λ . The true number of topics is $r = 4$. The top row, (a–c), depicts the training errors, and the bottom row, (d–f), depicts the testing errors. The first column, (a,d), is for a low rank approximation $r = 3$, the second column, (b,e), is for an approximation of correct rank $r = 4$, and the third column, (c,f), is for when the number of topics used $r = 5$ is larger than the true number of topics. For each λ , η , and r , fifty trials were run, and the regression errors corresponding to the best overall objective function $F^{(\lambda)}$ were recorded. Points with $\lambda > 0$ for which the regression error exceeds 1.5 times the regression error at $\lambda = 0$ are not displayed. The dashed vertical line is the transition point between a linear and logarithmic x -scale.

5. Rate My Professors Dataset

Here we study our method on real data [15] coming from Rate My Professors where for each instructor in the dataset, all corresponding student comments are combined to generate a written narrative and we have the instructor’s rating (mean of all responses).

5.1. Pre-Processing

The corpus was first processed via term frequency–inverse document frequency (TF-IDF) [25] with the TfidfVectorizer class in Python’s scikit learn package [26]. We used arguments `min_df=0.01`, `max_df=0.15`, `stop_words='english'`, `norm='l1'`, `lowercase=True`. We found the ratings were not balanced: there were 57 on the interval [1, 2), 235 on the interval [2, 3), 494 on the interval [3, 4), and 629 on the interval [4, 5]. To balance the dataset, we extracted only a random subset of 57 reviews in each interval (all ratings on [1, 2) were used). Overall, we obtained a corpus matrix X that was 228×1635 . The open right-end of the intervals ensures data are not duplicated.

5.2. Choice of Topic Number and Regression Weight

We did not know the true number of topics in the dataset and chose topics of $r = 1, 3, 5, 7, 9$, and 11, with $\lambda \in \{0\} \cup \{10^{2i/3} | i \in [-12, 0] \cap \mathbb{Z}\}$. We present the results for 11 topics that gave the best results. See Figure 4. We note that, for large enough λ , the testing error outperforms the testing error for $\lambda = 0$. The optimal point was at $\lambda = 10^{-2/3} \approx 0.215$.

We comment that it is generally difficult to know precisely where the testing error will be minimized, only that, based on observations of the synthetic data, the testing error is often better than the $\lambda = 0$ case after the training error has dropped. We speculate that the level of noise in this dataset results in the testing errors not dropping below ≈ 0.75 .

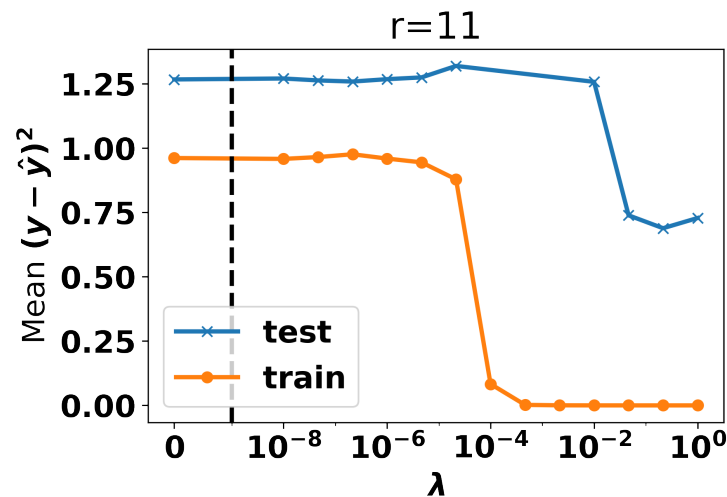


Figure 4. Errors in training and validation on Rate My Professor dataset with $r = 11$ topics. Points with $\lambda > 0$ for which the regression error exceeds 1.5 times the regression error at $\lambda = 0$ are not displayed. The dashed vertical line is the transition point between a linear and logarithmic x-scale.

5.3. Prediction

We examine the rating prediction by plotting histograms of predicted ratings where the true ratings were in $[1, 2]$, $[2, 3]$, $[3, 4]$, and $[4, 5]$ —the closed intervals are used here. Figure 5 depicts these histograms along with the mean predicted rating and true rating. The predictions are often within range, and the mean predicted values are very close to the true means over each interval. We can also see the general predictive strength in the scatterplot of actual vs. predicted ratings in Figure 6.

These results suggest the model is able to identify topics and associated θ -weights so as to generate predictions that are consistent with true ratings. For example, in the case where ratings are in $[1, 2]$, we see the peak of the predictions is around 2, not exceeding 4, with some predictions as low as -2 ; then, in the case of ratings in $[4, 5]$, the model peaks around 3.5 and makes some predictions above 7. There is a clear capacity for the topics to shift the predictions.

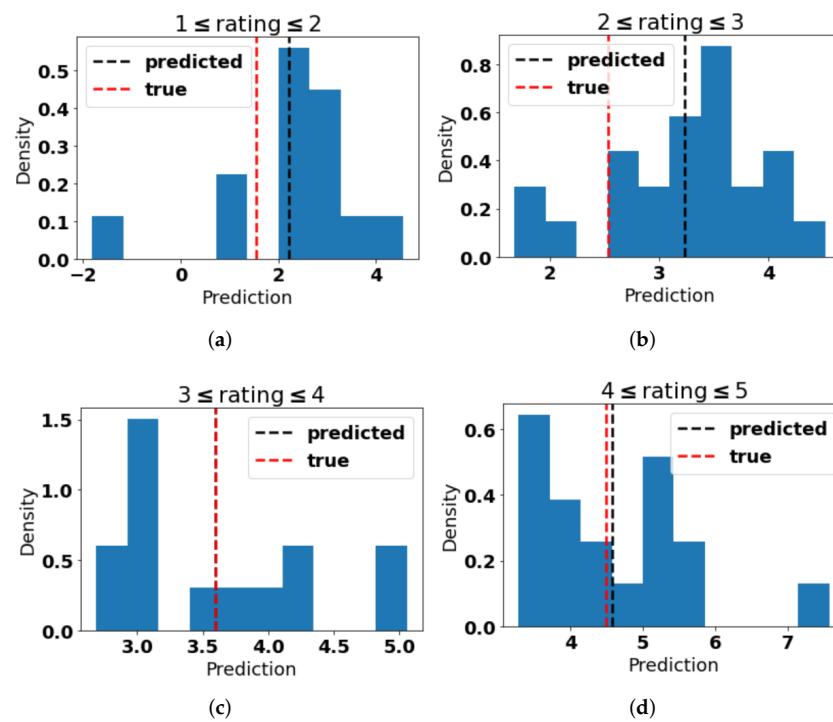


Figure 5. Histograms of the predicted mean rating for various ranges of true ratings ($[1, 2]$ in (a), $[2, 3]$ in (b), $[3, 4]$ in (c), and $[4, 5]$ in (d)). The vertical dashed lines represent the mean values. The predicted and true means are as follows: 2.206 and 1.543 for ratings in $[1, 2]$, 3.233 and 2.529 for ratings in $[2, 3]$, 3.594 and 3.593 for ratings in $[3, 4]$ (the lines are indistinguishable), and 4.576 and 4.494 for ratings in $[4, 5]$.

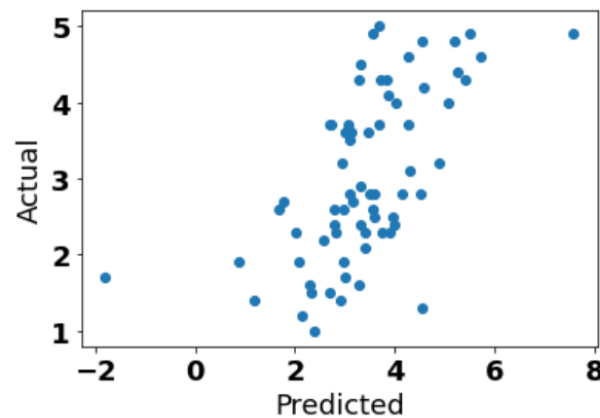


Figure 6. Scatterplot of Rate My Professor ratings: actual mean rating vs. predicted mean rating.

5.4. Topics Identified

It is important that the method not only have predictive power, but also produce interpretable topics. We now look at the 11 topics with their associated θ -weights. We find

$$\theta = (2.39909812, 2.82948873, -2.21028471, 1.83876976, -4.77504984, \\ -3.86467795, 3.46353642, 0.03914383, 3.26619842, -5.51595505, \\ 4.15317532, 3.90733652)^T.$$

Note that $\theta_1 \approx 2.4$ suggests that for a set of reviews with no topics, the average rating would be around 2.4—this suggests it is the presence of positive/negative topics that raise/lower the rating.

In Figures 7 and 8, we plot the words in the topics associated with positive and negative ratings. The topics are interpretable. For positive topics, we find Topic 10 (suggestive of extra credit), Topic 6 (suggestive of being inspiring), Topic 8 (suggestive of being approachable/brilliant), and Topic 11 (suggestive of being nice/enjoyable class) and words such as “recommend” in a couple of them. A few words seem out of place, such as “hate” in Topic 11, but that can be explained by some positive reviews having phrases such as “i hated chemistry in high school and after taking her class i don t [sic] hate chem as much.” Among the negative topics, we see Topic 4 (being horrible) and Topic 5 (being unfair).

As a whole, the topics are consistent with intuitive notions of what would be associated with higher or lower ratings. It is also interesting to look at the θ -topic weights quantitatively. For example, Topic 2 (mentioning rants and sarcasm) and Topic 4 (suggestive of being harder and failing students) contribute negatively to the score, but being a harder teacher seems to contribute more negatively to the rating than ranting. To elaborate more: because the row sums of the corpus matrix X and those of the topic matrix H sum to 1, we have that the sum of the topic weights for each document are approximately 1 as per Remark 1. All topics exist on the “same scale” within a document (roughly in $[0, 1]$) and must roughly sum to 1; therefore, the sizes of the weights in θ for each topic can be ordered by their positive/negative contributions.

The fact that instructor difficulty has a negative effect on rating and the easiness has a positive effect has been found in another study [27] that looked at Rate My Professors data on instructor easiness and overall quality ratings. It is also interesting that many of the “dimensions” detected through our study, such as being approachable/nice (niceness), being hard/easy (difficulty), and being inspiring, were dimensions naturally identified by other scholars [28] who analyzed Rate My Professors data by hand through reading comments and classifying key phrases within the comments. In this latter study, however, each dimension could be positive or negative, depending on how it was used.

| | | | |
|----------------------------|----------------------------|---------------------------|---------------------------|
| Topic 10, weight=4.153175: | Topic 11, weight=3.907337: | Topic 6, weight=3.463536: | Topic 8, weight=3.266198: |
| math 0.009225 | nicest 0.173561 | finance 0.301495 | laid 0.178756 |
| extra 0.006234 | hate 0.147343 | curriculum 0.072583 | fan 0.132521 |
| grading 0.004883 | high 0.133061 | master 0.071030 | state 0.114593 |
| credit 0.004513 | write 0.118530 | engineering 0.069513 | lol 0.107075 |
| favorite 0.004159 | enjoyed 0.118439 | inspirational 0.065388 | type 0.096690 |
| smart 0.004119 | school 0.116725 | called 0.065318 | mr 0.084134 |
| stats 0.003983 | complaint 0.006842 | manageable 0.062980 | thing 0.065279 |
| papers 0.003965 | everybody 0.005838 | theory 0.057255 | approachable 0.013363 |
| prepared 0.003939 | university 0.005283 | thats 0.054685 | university 0.012312 |
| asks 0.003719 | anymore 0.004907 | mind 0.045145 | brilliant 0.011068 |
| Topic 1, weight=2.829489: | Topic 3, weight=1.83877: | Topic 7, weight=0.039144: | |
| education 0.217164 | works 0.256413 | active 0.228253 | |
| entire 0.173686 | materials 0.147074 | fantastic 0.159382 | |
| instructor 0.168750 | keeps 0.118755 | field 0.156974 | |
| worst 0.139742 | instructor 0.107028 | looking 0.140967 | |
| life 0.137650 | tries 0.105110 | thought 0.119783 | |
| advice 0.006089 | doing 0.095939 | isn 0.109629 | |
| wall 0.005925 | stats 0.007298 | women 0.002632 | |
| recommended 0.004028 | recommended 0.005725 | lady 0.002310 | |
| enjoy 0.003364 | advice 0.004555 | cared 0.001618 | |
| elementary 0.003125 | project 0.002818 | mile 0.001616 | |

Figure 7. Topics with positive θ -weights. The θ -weight is given as the topic weight. The strength of each word is given numerically beside each of the top 10 words.

| | | | |
|----------------------------|---------------------------|----------------------------|----------------------------|
| Topic 9, weight=-5.515955: | Topic 4, weight=-4.77505: | Topic 5, weight=-3.864678: | Topic 2, weight=-2.210285: |
| stated 0.127021 | horrible 0.017942 | unfair 0.343236 | pick 0.518028 |
| according 0.125822 | worst 0.013849 | poor 0.341265 | chance 0.453842 |
| limited 0.125659 | usually 0.008699 | plus 0.291102 | sarcastic 0.001933 |
| criteria 0.108548 | doesnt 0.008120 | online 0.000877 | rants 0.001152 |
| participation 0.002192 | question 0.007518 | lose 0.000871 | nightmare 0.001120 |
| late 0.000243 | harder 0.007443 | excuse 0.000860 | format 0.001022 |
| required 0.076412 | failed 0.007395 | rd 0.000846 | middle 0.001012 |
| discussions 0.073464 | chapters 0.007029 | support 0.000715 | morning 0.000993 |
| avoid 0.008943 | math 0.006914 | john 0.000696 | spanish 0.000937 |
| online 0.001147 | gave 0.006894 | correctly 0.000688 | smartest 0.000876 |

Figure 8. Topics with negative θ -weights. The θ -weight is given as the topic weight. The strength of each word is given numerically beside each of the top 10 words.

6. Conclusions and Future Work

We have developed CSSNMF as a means to combine NMF with regression on a continuous response variable. We accomplished this by minimizing an objective function that combines an NMF error with a weighted regression error. We have shown that the regression error in training is weakly decreasing with the regression error weight and that, in practical applications, the error strictly decreases. The topics identified can outperform the quantitative accuracy of topics formed through NMF alone while retaining a high degree of interpretability (as found with real data). The method is robust to noise and tends to perform best on new data when there is a dimensionality reduction, i.e., when the number of topics fit for is fewer than the true number of topics.

We expect that our methods can be applied to very large datasets as our algorithmic steps involve hierarchical alternating nonnegative least squares [16], which could even be handled in parallel [29] (with or without parallel nonnegative least squares), and a least squares problem (Equation (12)) with a small $r \times r$ matrix to invert.

Although our analysis focused on the case of linear regression, incorporating nonlinearities would be of interest. We also noted the challenge in choosing the appropriate λ given only training data. A more theoretical understanding of when testing errors drop substantially could be explored, but this may be dataset-specific.

Author Contributions: Conceptualization, M.R.L. and D.N.; methodology, M.R.L.; software, M.R.L.; validation, M.R.L.; formal analysis, M.R.L.; investigation, X.D., F.L. and A.S.; data curation, M.R.L. and D.N.; writing—original draft preparation, M.R.L.; writing—review and editing, X.D., F.L., A.S. and D.N.; visualization, M.R.L.; supervision, M.R.L.; project administration, M.R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used have been cited and are publicly available. Our code is available at <https://bitbucket.org/3k1m/cssnmf/src/master/>, accessed on 21 February 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, Y.; Zhang, H.; Liu, R.; Ye, Z.; Lin, J. Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowl.-Based Syst.* **2019**, *163*, 1–13. [CrossRef]
- Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef] [PubMed]
- Lao, H.; Zhang, X. Regression and Classification of Alzheimer’s Disease Diagnosis Using NMF-TDNet Features From 3D Brain MR Image. *IEEE J. Biomed. Health Inform.* **2021**, *26*, 1103–1115. [CrossRef] [PubMed]
- Lai, Y.; Hayashida, M.; Akutsu, T. Survival analysis by penalized regression and matrix factorization. *Sci. World J.* **2013**, *2013*. [CrossRef]
- Stewart, G.W. On the early history of the singular value decomposition. *SIAM Rev.* **1993**, *35*, 551–566. [CrossRef]
- Shahnaz, F.; Berry, M.W.; Pauca, V.P.; Plemmons, R.J. Document clustering using nonnegative matrix factorization. *Inf. Process. Manag.* **2006**, *42*, 373–386. [CrossRef]
- Joyce, J.M. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 720–722.
- Marler, R.T.; Arora, J.S. The weighted sum method for multi-objective optimization: New insights. *Struct. Multidiscip. Optim.* **2010**, *41*, 853–862. [CrossRef]
- Freijeiro-González, L.; Febrero-Bande, M.; González-Manteiga, W. A critical review of LASSO and its derivatives for variable selection under dependence among covariates. *Int. Stat. Rev.* **2022**, *90*, 118–145. [CrossRef]
- Austin, W.; Anderson, D.; Ghosh, J. Fully supervised non-negative matrix factorization for feature extraction. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 5772–5775.
- Zhu, W.; Yan, Y. Joint linear regression and nonnegative matrix factorization based on self-organized graph for image clustering and classification. *IEEE Access* **2018**, *6*, 38820–38834. [CrossRef]
- Haddock, J.; Kassab, L.; Li, S.; Kryshchenko, A.; Grotheer, R.; Sizikova, E.; Wang, C.; Merkh, T.; Madushani, R.; Ahn, M.; et al. Semi-supervised Nonnegative Matrix Factorization for Document Classification. In Proceedings of the 2021 55th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 31 October–3 November 2021; pp. 1355–1360.

13. Li, P.; Tseng, C.; Zheng, Y.; Chew, J.A.; Huang, L.; Jarman, B.; Needell, D. Guided Semi-Supervised Non-negative Matrix Factorization on Legal Documents. *Algorithms* **2022**, *15*, 136. [CrossRef]
14. Rate My Professors. Available online: <https://www.ratemyprofessors.com/> (accessed on 17 February 2023).
15. He, J. Big Data Set from RateMyProfessor.com for Professors' Teaching Evaluation. 2020. Available online: <https://doi.org/10.17632/fvtfjyvw7d.2> (accessed on 21 February 2023).
16. Kim, H.; Park, H. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM J. Matrix Anal. Appl.* **2008**, *30*, 713–730. [CrossRef]
17. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*.
18. Klein, C.A.; Huang, C.H. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Syst. Man Cybern.* **1983**, *2*, 245–250. [CrossRef]
19. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
20. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
21. scipy.optimize.nnls. Available online: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.nnls.html> (accessed on 17 February 2023).
22. Bro, R.; De Jong, S. A fast non-negativity-constrained least squares algorithm. *J. Chemom. J. Chemom. Soc.* **1997**, *11*, 393–401. [CrossRef]
23. Luo, Y.; Duraiswami, R. Efficient parallel nonnegative least squares on multicore architectures. *SIAM J. Sci. Comput.* **2011**, *33*, 2848–2863. [CrossRef]
24. Berry, M.W.; Browne, M.; Langville, A.N.; Pauca, V.P.; Plemmons, R.J. Algorithms and applications for approximate nonnegative matrix factorization. *Comput. Stat. Data Anal.* **2007**, *52*, 155–173. [CrossRef]
25. Joachims, T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Technical Report, Carnegie-Mellon Univ Pittsburgh pa Dept of Computer Science. 1996. Available online: <https://apps.dtic.mil/sti/citations/ADA307731> (accessed on 21 February 2023).
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Bleske-Rechek, A.; Fritsch, A. Student Consensus on RateMyProfessors Com. *Pract. Assess. Res. Eval.* **2011**, *16*, 18.
28. Hartman, K.B.; Hunt, J.B. What ratemyprofessors. com reveals about how and why students evaluate their professors: A glimpse into the student mind-set. *Mark. Educ. Rev.* **2013**, *23*, 151–162.
29. Moon, G.E.; Ellis, J.A.; Sukumaran-Rajam, A.; Parthasarathy, S.; Sadayappan, P. ALO-NMF: Accelerated locality-optimized non-negative matrix factorization. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 1758–1767.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.