

Two to Five Truths in Non-Negative Matrix Factorization

John M. Conroy^{1*}, Neil P Molino^{1*}, Brian Baughman², Rod Gomez², Ryan Kaliszewski² and Nicholas A. Lines^{2*}

^{1*}Center for Computing Sciences, Institute for Defense Analyses,
17100 Science Drive, Bowie, 20715, Maryland, USA.

^{2*}United States Department of Defense.

*Corresponding author(s). E-mail(s): conroy@super.org;
npmolin@super.org; nicholasalines@gmail.com;

Abstract

In this paper we explore the role of matrix scaling on a matrix of counts when building a topic model using non-negative matrix factorization. We present a scaling inspired by the normalized Laplacian (NL) for graphs that can greatly improve the quality of a non-negative matrix factorization. The results parallel those in the spectral graph clustering work of [1], where the authors proved adjacency spectral embedding (ASE) spectral clustering was more likely to discover core-periphery partitions and Laplacian Spectral Embedding (LSE) was more likely to discover affinity partitions. In text analysis non-negative matrix factorization (NMF) is typically used on a matrix of co-occurrence “contexts” and “terms” counts. The matrix scaling inspired by LSE gives significant improvement for text topic models in a variety of datasets. We illustrate the dramatic difference a matrix scalings in NMF can greatly improve the quality of a topic model on three datasets where human annotation is available. Using the adjusted Rand index (ARI), a measure cluster similarity we see an increase of 50% for Twitter data and over 200% for a newsgroup dataset versus using counts, which is the analogue of ASE. For clean data, such as those from the Document Understanding Conference, NL gives over 40% improvement over ASE. We conclude with some analysis of this phenomenon and some connections of this scaling with other matrix scaling methods.

Keywords: Laplacian, Topic Modelling, NMF

Introduction

In their paper [1] the authors provide a clear and concise demonstration of the “two-truths” in spectral graph clustering. Their results prove that the choice of the first step—spectral embedding of either Laplacian spectral embedding (LSE) or adjacency spectral embedding (ASE) will identify different underlying structures, when present in a graph. The results, which were empirically observed at first, were made precise using Chernoff information and a stochastic a block model of the underlying graphs.

The two-truth property was observed empirically while computing on connectome models of the human brain [1]. Figure 1, which is drawn using a 1 million vertex graph model of the brain, shows the two canonical partitions of the brain by either hemisphere (left/right) or grey matter/white matter. In graph theoretic terms the former partition is an “affinity” partition and the latter “core-periphery.” Prior to [1] it was thought ASE and LSE would give comparable partitions. Figure 2 gives an example of both the Laplacian spectral embedding and adjacency spectral embeddings. The first exposes the hemispherical (left vs. right) (an affinity partition) bisection and the latter a partition which is strongly correlated with grey matter vs white matter, which in this connectome model is a core-periphery partition.

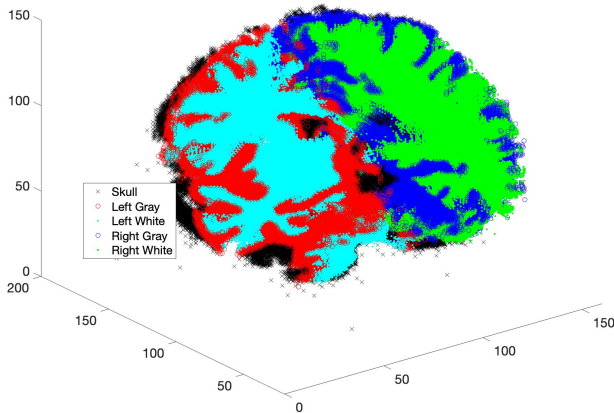


Fig. 1 Ground truth partitions of a connectome model of a human brain.

The advantage of using a non-negative matrix factorization for data analysis was first noted by [2] and five years later its power was demonstrated in the Nature article [3]. In computational linguistics a matrix of co-occurrence of terms with their *context* is often employed. A context is the set of one or more terms that precede or follow a word in a collection of text [4]. Depending on the application a context could be a set of two or more consecutive words, a sentence, or one or more documents. (In this work, our main examples will

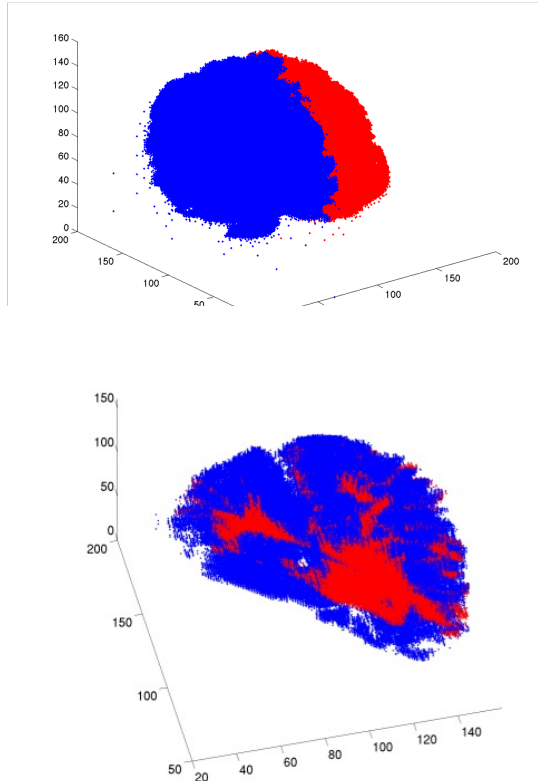


Fig. 2 Laplacian Spectral and Adjacency spectral embeddings bisections of a connectome model of a human brain.

be in the latter case of one or more documents as the *context* in our models.) The spectral methods applied in the text application uncover topic models, clusters of words (or more generally terms) that are related for LSE for a set of documents and ASE will tend to find “key concepts” versus “less important” information (peripheral ideas). An example using a spectral method in computation linguistic to find the key sentences was successfully used by [5].

In this current work we seek to explore empirically to what extent the two-truth phenomenon extends to non-negative matrices, which arise commonly in the analysis of text data. Unlike the graph and their corresponding 0-1 symmetric adjacency matrices from the connectome problem, text models generally consist of integer count values and are weighted bipartite graphs as opposed to simple graphs. As the entries are counts, a non-negative factorization (NMF) is often used in text analysis. Analogous to [5] the authors in the paper [6] employed a NMF on an incidence term-sentence matrix to find key parts of a scientific document and citation sentences from papers that reference it. We define a normalized Laplacian scaling of such a matrix as well as other natural

scalings in the next section. We proceed with computational experiments and discuss some theoretical considerations as related work.

1 Bipartite Laplacian and Other Matrix Scalings

The motivation of the matrix scaling comes from a scaling of a symmetric matrix A , the adjacency matrix of a graph. Our application is a weighted bipartite graph. So, we adapt the technique with a block matrix construction. For a matrix M with all non-negative entries, representing our document-term matrix, construct the specific matrix, A , as

$$A = \begin{bmatrix} 0 & M \\ M^T & 0 \end{bmatrix},$$

and we denote n as the number of documents and m the number of terms, giving the number of rows and columns of M , respectively.

Let $D_{r,A} = \text{diag}(A\mathbf{1})$, where $\mathbf{1}$ is the all ones vector, i.e., $D_{r,A}$ is the diagonal matrix of row sums of A . We also note that we can use this to define the column sum diagonal as $D_{c,A} = D_{r,A^T}$

The matrix A may be viewed as the adjacency matrix of a weighted bipartite graph, where we connect every word to each document by an edge weighted by the number of times that word appears in that document. With a weighted bipartite graph model, we can consider the usual variants of the graph Laplacian.

Markov: $A_{\text{Markov}} = D_{r,A}^{-1}A$

Laplacian: $L = D_{r,A} - A$

Random Walk Normalized Laplacian (RWNL): $L_{\text{rwnl}} = I - A_{\text{Markov}}$

Normalized Laplacian (NL): $L_{\text{nl}} = I - D_{r,A}^{-\frac{1}{2}}AD_{r,A}^{-\frac{1}{2}} = I - A_{\text{nl}}$

The normalized Laplacian scaling for the matrix M can be expressed as

$$M_{\text{nl}} = D_{r,M}^{-\frac{1}{2}}MD_{c,M}^{-\frac{1}{2}}. \quad (1)$$

In addition to this matrix scaling the algebra suggests several other representations. In particular, normalizing within the documents, i.e., row scaling (RS), (equation 2), normalizing term counts across documents, i.e. column scaling (CS), (equation 2), and normalizing the rows and columns independently, which is the exponential of point-wise mutual information (PWMI), (equation 4).

$$M_r = D_{r,M}^{-1}M \quad (2)$$

$$M_c = MD_{c,M}^{-1} \quad (3)$$

$$M_{rc} = D_{r,M}^{-1}MD_{r,M}^{-1} \quad (4)$$

These four scaling in addition to the matrix M will be used as alternative matrices in our NMF factorization.

Recall that we aim to consider non-negative matrix factorization of scalings of the weighted bipartite graph. Let us posit the model [7] representation of the matrix M

$$M = WH + E$$

where W is a m by k non-negative matrix, H is a k by m non-negative matrix, and E is a matrix of residual errors. The low-rank non-negative factorization is frequently interpreted as a topic model of k topics and the factors W and H are proportional to counts. In this way, the row normalized version of W and H are thought of as multinomial distributions expressing each document as a mixture of k topics. Each topic is a multinomial distribution over the words in the vocabulary. The mixing weights over the topics for each document are used to “assign” (associate) each document to a primary topic. The resulting factors \tilde{W} and \tilde{H} are then scaled by the inverse of the diagonal scaling applied a priori. Thus, the following pre and post scaling is done in the proposed diagonally scaled NMF.

1. *Pre-scaling*: Let D_r and D_c be the diagonal scaling matrices for the rows and columns of our data matrix M ; we then form

$$\tilde{M} = D_r^{-1} M D_c^{-1}$$

2. *Factorization*: Compute a non-negative matrix factors \tilde{W} and \tilde{H}

$$\tilde{M} = \tilde{W} \tilde{H} + \tilde{E}.$$

3. *Post-scaling*: Compute

$$\begin{aligned} W &= D_r \tilde{W} \\ H &= \tilde{H} D_c \end{aligned}$$

2 Computational Results

In this section we demonstrate the relative performance of the the five matrix scalings when using a NMF to cluster a set of documents into topics. We first introduce the datasets, then discuss how the data are processed, and finally present the evaluation measures and results.

2.1 Three Datasets of Varying Difficulty

We illustrate the performance of the method on three datasets, which vary from relatively “well-separated” topics to “well-mixed” topics. The datasets are:

1. Document Understanding Conference 2004 (DUC 2004) multidocument summarization data. These data are 500 newswire documents which were selected to answer query needs for 50 topics. The documents were carefully

chosen. First, a search engine was used to retrieve documents for each topic based on a human query. For each topic, a human selected the 10 most relevant documents. This data set is known to be of high quality and is one of many widely used in the summarization literature.¹

2. The 20 newsgroups training data set, which consists of approximately 11K posts to one of 20 newsgroups. Each newsgroup has roughly between 400 and 600 posts. We note that the newsgroups data topics are more broadly defined than the DUC 2004 documents and will have the occasional post which may be off the group’s topic.
3. Russian Troll Twitter dataset. The English port of these data were labeled by the Clemson University researchers [8] into one of eight troll types. ‘Commercial,’ ‘Fearmonger,’ ‘HashtagGamer,’ ‘LeftTroll,’ ‘News-Feed,’ ‘NonEnglish,’ ‘RightTroll,’ and ‘Unknown.’ Limiting the focus to the English tweets, there are 1648 trolls which have been assigned one of the 8 labels.²

These labels define how the Twitter actors *behaved*. Each actor’s behavior would span multiple *topics* in the normal notion of a text topic model. We include these data as a challenge dataset and also as a dataset which will have text properties not generally found in newswire documents or newsgroup posts.

Here we apply the five matrix scalings:

1. *None*: the original counts,
2. *CS* column scaling (CS),
3. *RS* row scaling,
4. *PWMI* pointwise mutual information, and
5. *NL* normalized Laplacian (NL).

As the matrices are document-term matrices, RS scaling turns each row into the maximum likelihood multinomial distribution for a document’s vocabulary. Similarly, CS estimates the term distributions across the contexts (a.k.a. documents in this case). PWMI scales the counts by the row and column marginals, while NL scales them by the square roots of these marginals. To the extent that the spectral result of [1] carries over to NMF applied to text we would expect NL to be best at recovering latent topics.

2.2 From Data to Matrices

In this subsection we describe how the data are processed to create context-term matrices upon which we perform the five variants of non-negative matrix factorization.

Before we model the documents as a context-term matrix of counts, we must first define the notion of a *context*. The down stream task, in our case, clustering, defines how the data are separated into contexts. For the newswire

¹See <https://www-nlpir.nist.gov/projects/duc/data.html> for the DUC 2004 and other DUC data. Also, see <https://tac.nist.gov/data> for similar data and their descriptions.

²<https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/>

dataset, a context is simply a single newspaper article. Similarly, a context for the newsgroup data is an individual post. For the Russian Troll tweets, the task is to cluster individual users, which are known *trolls*, actors who by nature of some their tweets have malicious intents.³ So, here the natural notion of context is the collection of tweets from a given Twitter account.

Once the contexts are identified they are then broken into parts, which we call *terms*. Notionally, a term could be thought of as a word, but more formally, it is a function which maps documents to a vector of counts of fixed length, n , the number terms. The term definition is data dependent and is specified by a series of three parts; more formally the function can be viewed as composition of three functions. From the introduction of the vector space model [9] nearly 50 years ago it was realized that words (tokens) which occur either *too frequently* or *too rarely* do not have discriminating power and are best removed from the index. We employ information theoretic methods to remove such common and rare tokens from the index.

1. *Mapping from text to terms:* We employ two approaches to break documents into terms: first a function provided by the python module `sklearn.feature_extraction.text`⁴ and second a tokenization method popular with neural net models, `sentencepiece`. `sentencepiece` is used in Text to Text Transformer Transfer learning (T5)[10] and code supported by `huggingface.com`.⁵
 - (a) `CountVectorizer` by default employs a simple regular expression ignoring punctuation and splitting the text into tokens based on white space. In our experiments we use `CountVectorizer`'s default options.
 - (b) `sentencepiece` [11] is the tokenization method in T5 and it employs a unigram language model to find a set of tokens (consecutive bytes in the data) of size N , an input parameter, which maximizes the probability of the text. We use the pre-trained model provided, which converts text into a set of tokens (vocabulary) of approximately size 30 thousand.
2. *Removing Common Tokens:* The procedure for removing common tokens is often done with a fixed list, commonly called a *stop word list* of the most common words in the language. The approach here is algorithmic and data dependent. The choice of what tokens are removed is based on a likelihood score. The tokens are sorted by frequency and a model is posed that assumes there is a change point between the commonly occurring tokens and the tokens specific to the given data. Given a count threshold, the counts are divided into two populations. For each population the first two moments are computed and a likelihood score is computed assuming normality of the two populations. The threshold giving the maximum likelihood is then chosen to divide the populations of tokens. Tokens with the higher counts are then removed.

³“In Internet slang, a troll is a person who posts inflammatory, insincere, digressive, extraneous, or off-topic messages in an online community ...” (See https://en.wikipedia.org/wiki/Internet_troll)

⁴https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text

⁵https://huggingface.co/docs/transformers/model_doc/t5

3. *Removing Rare Tokens*: The procedure for removing rare tokens is quite basic, but like the common token removal, data dependent. Token counts are sorted and a cumulative sum of the counts is computed and a threshold is chosen so as to keep 99% of the total of the counts.

2.3 Clustering with an NMF and Evaluating Performance

To cluster the contexts we compute a NMF for the data for a given rank, k . In the matrix factorization WH each row of the matrix W gives non-negative values which are the relative strength of each “topic.” We assign each document to the topic for which it has the largest weight.⁶

As the quality of the factorization may vary based on the choice of k we compute for a range of values. Use use a likelihood method due to Zhu and Ghodsie (ZG) [12] to estimate the number of topics and implemented by **graspy**⁷. The singular values of the matrix M are computed and the “second elbow,” as discovered by the likelihood method. NMF factorization using all five variants of the dimension given by this elbow as well as 10 to the left and 10 to right of the chosen range of dimensions evaluated.

The quality levels of the resulting clusters are then evaluated using the Adjusted Rand Index (ARI) [13] [14] [15]. The Rand index allows for comparing two clustering methods, each of which may have a different numbers of clusters. The Rand index [13] is defined simply as the fraction of times the two clustering methods agree whether pairs of items are in the same cluster or not. It is based on all pairs of items. There are $\binom{m}{2}$ of these, where m is the number of the items being clustered. The adjustment part, as proposed by [14] and further studied by [15], subtracts off the expected Rand index for a random partitioning of size k , where k is the number of clusters found in the model. The values of an ARI are bounded above by 1, but can be negative when a clustering performs worse than random.

Figures 3 through 5 give the ARI for the five scalings of NMF for the range of values around the maximum likelihood estimated dimension. Here we see rather strong performance with an ARI of about 0.7 and the NL scaling gives slightly better results than the second best scaling of RS, which normalizes within the documents.

The 20 newsgroup data are overall more challenging to cluster, but here the NL scaling gives a sizable improvement over the alternative methods.

Finally, for the Twitter data, the most challenging dataset studied, the performance as measured by ARI is lower still, but with NL giving better results.

Each of the above experiments used **CountVectorizer** to tokenize the data. To illustrate that the results can vary substantially based the tokenization we present the last dataset again comparing **CountVectorizer** with the T5’s **sentencepiece**, the unigram language model. Figure 6 gives these results,

⁶We observe that with this simple approach the post scaling has no affect on the clustering, i.e. using \tilde{W} instead of W would yield the same clusters.

⁷<https://github.com/bdpedigo/graspy>

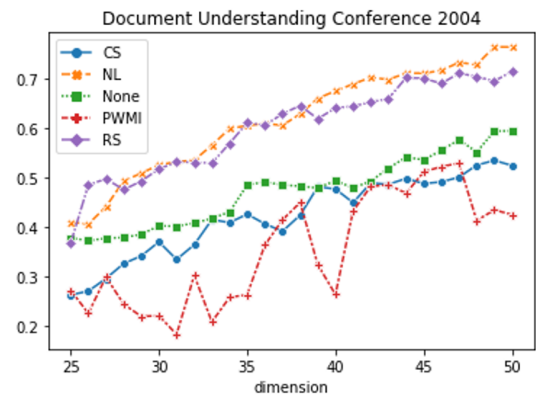


Fig. 3 Adjusted Rand Index for 5 matrix scalings: DUC 2004 Dataset.

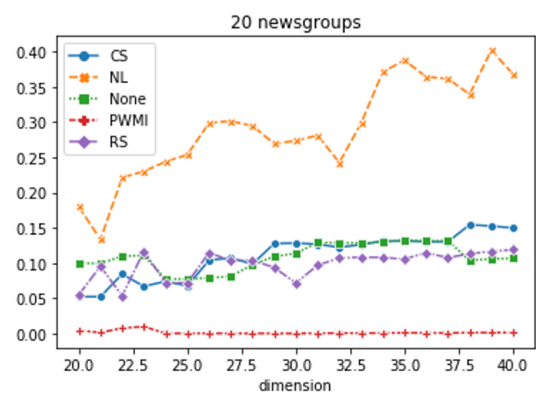


Fig. 4 Adjusted Rand Index for 5 matrix scalings:20 News Groups Dataset.

where we see both original scaling (Counts) and the normalized Laplacian (NL) improve significantly and perform comparably.⁸

3 Discussion

Here we point out that the computations that we are doing are ultimately based on on `sklearn`'s routine for non-negative matrix factorization, `NMF`. We mostly use the default parameters, but it is worth discussing a few of the major ones and the impact that they have. Specifically, we note that by default, and hence in our calculations there is no explicitly regularization. The package allows for both L_1 and L_2 or a combination of them.

⁸Using the T5 tokenizer on the DUC 2004 and newsgroups data give somewhat lower performing clustering, but the relative performance of the scalings remains about the same.

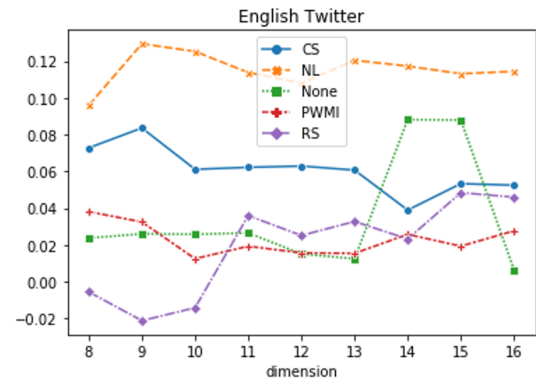


Fig. 5 Adjusted Rand Index for 5 matrix scalings: Russian Twitter Trolls.

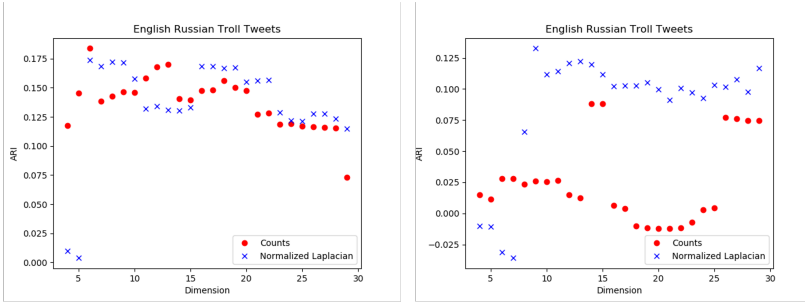


Fig. 6 Comparison of `sentencepiece` vs `CountVectorizer` Tokenization: Russian Twitter Trolls.

The second important parameter we will spend some time investigating is that of the loss function itself. In any case, the goal is to minimize a component-wise loss function, $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.

$$\min \sum_{i,j}^n L(M_{ij}, (WH)_{ij})$$

Indeed, we see in the screenshot 7 of `sklearn`'s documentation that there is an input to NMF called `beta_loss`. This can be passed a float or one of three special strings. The special strings are `frobenius`, `kullback-leibler`, or `itakuru-saito` which correspond to the values 2, 1, and 0 respectively. Their specific functional forms can be seen below.

$$L_{\text{Frob}}(x, y) = (x - y)^2$$

$$L_{\text{KL}}(x, y) = x \log \frac{x}{y}$$

$$L_{\text{IS}}(x, y) = \frac{x}{y} - \log \frac{x}{y} - 1$$

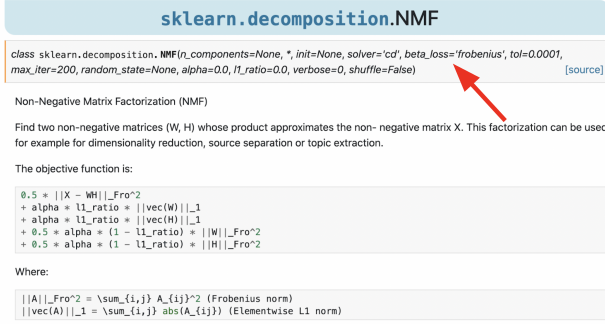


Fig. 7 Screenshot of documentation of `sklearn`'s NMF function.

The main point is that interaction of loss function and feature scaling is extremely important. Specifically, we note that when we scale both x and y by the same constant α the Frobenius loss is multiplied by α^2 , the Kullback-Leibler loss is multiplied by a single factor of α and the Itakuru-Saito loss is unaffected. I.e.,

$$\begin{aligned} L_{\text{Frob}}(\alpha x, \alpha y) &= \alpha^2 L_{\text{Frob}}(x, y) \\ L_{\text{KL}}(\alpha x, \alpha y) &= \alpha L_{\text{KL}}(x, y) \\ L_{\text{IS}}(\alpha x, \alpha y) &= L_{\text{IS}}(x, y) \end{aligned}$$

Since we are using the Frobenius norm as our loss in this setting, we see that the loss will be dominated by large values. The quadratic scaling of the penalty means that the NMF will work very hard to match the large terms in the summed loss function.

For ease of notation, we will look at what specifically happens to the entries in these scalings for the very simple two documents with two words each setting. That is, we will be looking at 2×2 matrices. We will think of documents as the rows and the terms index the columns.

3.1 Counts Matrix

When the matrix, A , is nothing more than the counts of words in each document, then as previously mentioned, the Frobenius norm will focus on the entries with larger values. Those will be ones that are either common words or those corresponding to longer documents. The algorithm will put much more emphasis on getting those entries correct. Obviously all of the entries here lie in \mathbb{N}

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

3.2 Row Scaling

When we consider the row scaling,

$$D_{r,A}^{-1}A = \begin{bmatrix} \frac{a}{a+b} & \frac{b}{a+b} \\ \frac{c}{c+d} & \frac{d}{c+d} \end{bmatrix},$$

then we find that the preference for long documents is neutralized. Common words, however, are still emphasized. Also, it is clear that all of the entries in this case lie in the unit interval, $[0, 1]$.

3.3 Column Scaling

The column scaling setting is the opposite as the preference for common words is neutralized, but the preference for long documents is not. Again, the entries here lie in $[0, 1]$.

$$AD_{c,A}^{-1} = \begin{bmatrix} \frac{a}{a+c} & \frac{b}{b+d} \\ \frac{c}{a+c} & \frac{d}{b+d} \end{bmatrix}$$

3.4 Normalized Laplacian Scaling

The Normalized Laplacian setting strikes a balance between the two. The preference for frequent words and for long documents are both partially neutralized. Again, the entries lie in the unit interval.

$$D_{r,A}^{-\frac{1}{2}}AD_{c,A}^{-\frac{1}{2}} = \begin{bmatrix} \frac{a}{\sqrt{(a+b)(a+c)}} & \frac{b}{\sqrt{(a+b)(b+d)}} \\ \frac{c}{\sqrt{(a+c)(c+d)}} & \frac{d}{\sqrt{(c+d)(b+d)}} \end{bmatrix}$$

3.5 Pointwise Mutual Information Scaling

The (exponentiated) point-wise mutual information scaling really came about from a very nice theoretical analysis done in [16]. The authors looked at the critical points of `word2vec`'s objective function [17]. When that is achieved the embedding vectors form a low rank factorization of the pointwise mutual information matrix.

$$A_{\text{PwMI}} = \begin{bmatrix} \frac{a}{(a+b)(a+c)} & \frac{b}{(a+b)(b+d)} \\ \frac{c}{(a+c)(c+d)} & \frac{d}{(c+d)(b+d)} \end{bmatrix}$$

First of all the name comes from the fact that entries resemble the summand (or integrand) of the definition of the mutual information between two variables.

$$I(X; Y) = \sum_{x_i, y_j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

If we look inside the log, we find the ratio of a joint probability with the product of the marginals. In this way, it can be seen as a measure of how far

two random variables are from independent. In this text setting, it measures the deviation from the expected word count if we just took into account the overall word frequency and the length of the document. The calculation below normalizes the numerator and denominators in the matrix by n , the total number of words in the corpus. There, we can see that the entries are the maximum likelihood estimates of the probabilities given the corpus and match the form of the point-wise mutual information.

$$nD_{r,A}^{-1}AD_{c,A}^{-1} = n \begin{bmatrix} \frac{a}{(a+b)(a+c)} & \frac{b}{(a+b)(b+d)} \\ \frac{c}{(a+c)(c+d)} & \frac{d}{(c+d)(b+d)} \end{bmatrix} = \begin{bmatrix} \frac{\frac{a}{n}}{(\frac{a+b}{n})(\frac{a+c}{n})} & \frac{\frac{b}{n}}{(\frac{a+b}{n})(\frac{b+d}{n})} \\ \frac{\frac{c}{n}}{(\frac{a+c}{n})(\frac{c+d}{n})} & \frac{\frac{d}{n}}{(\frac{c+d}{n})(\frac{b+d}{n})} \end{bmatrix}$$

Perhaps a little less obvious is the relationship between the pointwise mutual information and the Normalized Laplacian formulation. The calculation below

$$\frac{P(X=1, Y=1)}{P(X=1)P(Y=1)} = \frac{\frac{a}{n}}{\frac{a+b}{n} \frac{a+c}{n}} = \frac{an}{(a+b)(a+c)} = \frac{\frac{a}{\sqrt{(a+b)(a+c)}}}{\sqrt{\frac{a+b}{n} \frac{a+c}{n}}} \quad (5)$$

looks at the top left entry of the matrix in particular

$$(A_{PwMI})_{11} = \frac{\frac{a}{n}}{\frac{a+b}{n} \frac{a+c}{n}}$$

and relates it to the top left entry of the Normalized Laplacian matrix

$$(A_{NL})_{11} = \frac{a}{\sqrt{(a+b)(a+c)}}.$$

In particular, we see that

$$(A_{PwMI})_{11} = \frac{(A_{NL})_{11}}{\sqrt{\frac{a+b}{n} \frac{a+c}{n}}}$$

We further note that the denominator of the right hand side in equation 5 is the geometric mean of the two marginals. It has the possibility of being small and so the entries in the pointwise mutual information matrix are not necessarily constrained to lie in the unit interval and, hence, are sensitive to the quadratic scaling of L_{Frob} discussed earlier. It reintroduces a bias towards rare words or short documents.

4 Related Work

Unlike, the singular value decomposition (SVD), the solution to the NMF problem is NP-hard [18]. Not only that, but it is ill-posed; there is often not a unique answer. For example $A = WH = WTT^{-1}H = (WT)(T^{-1}H) = \hat{W}\hat{H}$ for any invertible matrix T . This makes a similar analysis to that done in [1] less promising.

There are situations where NMF does indeed have a well-behaved unique solution. Donoho and Stodden gave a nice geometric characterization of one such case where the matrix is called separable [19]. This was further brought into the text domain in [20]. Here the separability is a relatively mild assumption. It corresponds to the presence of so-called ‘anchor’ words. An anchor word is one that is supported on one and only one topic. This is plausible for text data. For example, it is likely that the word ‘homeomorphism’ only really appears in a topic about mathematics or ‘EBITDA’ really only appears in a topic about finance. Under these assumptions, there are unique answers and efficient algorithms to find them. These algorithms are, however, sensitive to the condition number of the matrices involved. So, there is a line of work that aims to remedy these algorithms by preconditioning the matrix, M . For example, [7] uses semidefinite programming to compute a minimum volume ellipsoid. We conjecture that our Normalized Laplacian method is related to a relaxation of this problem where the ellipsoid is constrained to be axis aligned.

The authors [21] propose using diagonal row and column scaling of a matrix of counts, i.e., Sinkhorn balancing, when employing spectral clustering. The authors show that when a block structure of count matrix is present that the singular vectors of the balanced matrix exhibit a “square-wave” pattern, when reordered. They propose an algorithm using a few singular vectors to permute a matrix of counts into a block structure. A natural question is how to best extend such an approach to rectangular matrices. The authors are currently working on such an approach. Such approaches should also be studied for a non-negative factorization.

The authors [22] formulate a low-rank Sinkhorn algorithm for optimal transport. They in affect are computing a NMF of a doubly stochastic matrix.

5 Conclusions

In this paper we proposed a new family of non-negative matrix factorizations inspired by normalized Laplacian and spectral graph theory. Previous work in spectral graph theory suggests that the normalized Laplacian gives rise to graph partitions which are more likely to find affinity communities in a graph versus spectral partitions based on the adjacency matrix, which tend to recover core-periphery partitions. In the non-negative matrix applications, it suggested that perhaps a normalized Laplacian scaling of the the counts may give rise to better topic models. In this paper we gave strong evidence for the normalize Laplacian giving better topic models as illustrated in three text

clustering datasets. In addition, we found that the choice of tokenization can significantly affect the actions between the matrix scaling and related work.

References

- [1] Priebe, C.E., Park, Y., Vogelstein, J.T., Conroy, J.M., Lyzinski, V., Tang, M., Athreya, A., Cape, J., Bridgeford, E.: On a two-truths phenomenon in spectral graph clustering. *Proceedings of the National Academy of Sciences* **116**(13), 5995–6000 (2019) <https://arxiv.org/abs/https://www.pnas.org/doi/pdf/10.1073/pnas.1814462116>. <https://doi.org/10.1073/pnas.1814462116>
- [2] Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* **5**(2), 111–126 (1994) <https://arxiv.org/abs/https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.3170050203>. <https://doi.org/10.1002/env.3170050203>
- [3] Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. *Nature* **401**, 788–791 (1999)
- [4] Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**(null), 1137–1155 (2003)
- [5] Steinberger, J., Jezek, K.: Using latent semantic analysis in text summarization and summary evaluation. In: *Proceedings of the 7th International Conference ISIM* (2004)
- [6] Conroy, J.M., Davis, S.T.: Section mixture models for scientific document summarization. *Int. J. Digit. Libr.* **19**(2-3), 305–322 (2018). <https://doi.org/10.1007/s00799-017-0218-6>
- [7] Gillis, N., Vavasis, S.: Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization. *SIAM Journal on Optimization* **25** (2013). <https://doi.org/10.1137/130940670>
- [8] Linvill, D.L., Boatwright, B.C., Grant, W.J., Warren, P.L.: “the russians are hacking my brain!” investigating russia’s internet research agency twitter tactics during the 2016 united states presidential campaign. *Computers in Human Behavior* **99**, 292–300 (2019). <https://doi.org/10.1016/j.chb.2019.05.027>
- [9] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* **18**(11), 613–620 (1975). <https://doi.org/10.1145/361219.361220>
- [10] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M.,

- Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* **21**(140), 1–67 (2020)
- [11] Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (2018). <https://doi.org/10.18653/v1/D18-2012>. <https://aclanthology.org/D18-2012>
- [12] Zhu, M., Ghodsi, A.: Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Comput. Stat. Data Anal.* **51**(2), 918–930 (2006)
- [13] Rand, W.M.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**(336), 846–850 (1971) <https://arxiv.org/abs/https://www.tandfonline.com/doi/pdf/10.1080/01621459.1971.10482356>. <https://doi.org/10.1080/01621459.1971.10482356>
- [14] Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* **2**(1), 193–218 (1985). <https://doi.org/10.1007/BF01908075>
- [15] Nguyen, X.V., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010)
- [16] Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., ??? (2014). <https://proceedings.neurips.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf>
- [17] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 26. Curran Associates, Inc., ??? (2013). <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>
- [18] Vavasis, S.A.: On the complexity of nonnegative matrix factorization. *SIAM J. on Optimization* **20**(3), 1364–1377 (2009)
- [19] Donoho, D., Stodden, V.: When does non-negative matrix factorization give correct decomposition into parts? *Advances in neural information processing systems* **16** (2004)

- [20] Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., Zhu, M.: Learning topic models – provably and efficiently. *Commun. ACM* **61**(4), 85–93 (2018). <https://doi.org/10.1145/3186262>
- [21] Le Gorrec, L., Mouysset, S., Duff, I.S., Knight, P.A., Ruiz, D.: Uncovering Hidden Block Structure for Clustering. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2019)*. *Lecture Notes in Computer Science* book series, vol. 11906, pp. 140–155. Springer, Würzburg, Germany (2019). https://doi.org/10.1007/978-3-030-46150-8_9. <https://hal.archives-ouvertes.fr/hal-03003811>
- [22] Scetbon, M., Cuturi, M., Peyré, G.: Low-rank sinkhorn factorization. In: Meila, M., Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 139, pp. 9344–9354. PMLR, ??? (2021). <http://proceedings.mlr.press/v139/scetbon21a.html>