SIMPLEX DECOMPOSITIONS FOR REAL-VALUED DATASETS

Madhusudana Shashanka

Mars, Inc., 100 International Drive, Mount Olive NJ 07828

ABSTRACT

In this paper, we introduce the concept of *Simplex Decompositions* and present a new Semi-Nonnegative decomposition technique that works with real-valued datasets. The motivation stems from the limitations of topic models such as Probabilistic Latent Semantic Analysis (PLSA), that have found wide use in the analysis of non-negative data apart from text corpora such as images, audio spectra, gene array data among others. The goal of this paper is to remove the non-negativity requirement for datasets so that these models can work on datasets with both positive and negative entries. We start by showing that PLSA is equivalent to finding a set of components that define the corners of a simplex within which all datapoints lie. We formalize this intuition by introducing the notion of *Simplex Decompositions* - PLSA and extensions are specific examples - and generalize the idea to be applicable to arbitrary real datasets with both positive and negative entries. We present algorithms and illustrate the method with examples.

1. INTRODUCTION

The problem of analyzing non-negative data appears in many diverse fields such as computer vision, semantic analysis, analysis of audio spectra and gene expression analysis. The goal in such applications is to find suitable representations that make hidden structure in the data explicit. Methods such as Singular Value Decomposition (SVD), Principal Components Analysis (PCA), Independent Components Analysis (ICA) and Projection Pursuit, are not suitable for such data and techniques that exclusively deal with non-negative data have gained in popularity. These techniques can broadly be classified into linear-algebra inspired Non-negative Matrix Factorization (NMF) and its derivatives; and probabilistic topic models such as Probabilistic Latent Semantic Analysis (PLSA; [1]) and its extensions such as Latent Dirichlet Allocation [2] and Correlated Topic Models [3] among others.

These latter methods that work only on non-negative data implicitly impose non-negativity constraints on all the components that are extracted. More specifically, they explain the given non-negative data as a guaranteed *non-negative linear combination* of a set of non-negative “bases” that represent realistic “building blocks” for the data. The fact that co-efficients in the linear combination are non-negative implies that all “bases” can combine only additively without any cross-cancellations to approximate the input. This has intuitive appeal as bases can then be considered as “parts” that combine in different ways to give rise to the entire dataset.

Probabilistic topic models, in addition, impose another constraint whereby all extracted components have to be *multinomial distributions*. When viewed as matrix decompositions, it implies that the non-negative entries of basis-vectors and corresponding mixture-weight vectors should also sum to unity. Since the same set of basis vectors combine with various mixture weights that sum to 1, the model approximations to the data points can be considered as lying in a simplex defined by the basis vectors.

In this paper, we formalize this idea as *Simplex Decompositions* where the “parts-based” decomposition of the dataset has the property that basis vectors combine additively and correspond to the corners of a simplex surrounding the modeled data. We extend it to work on *arbitrary datasets with both positive and negative entries*. This is accomplished by a principled series of steps that transform the dataset with real entries into one with only non-negative entries so that topic models can then be applied on the transformed dataset. The results are then transformed back to the original data space. Such a decomposition of the real-valued dataset is only “semi-nonnegative,” with the basis vectors having real-valued entries while the mixture-weight vectors are constrained to non-negative entries that sum to 1. We christen this new family of techniques as *Real Topic Models*.

The paper is organized as follows. In Section 2, we describe the PLSA algorithm and introduce the idea of Simplex Decomposition. In Section 3, we propose a method to extend Simplex decompositions to datasets that can have both positive and negative entries. Section 4 provides a discussion of the algorithm and compares the method to related techniques. We conclude the paper in Section 5 with a brief summary and avenues for future work.

2. PRELIMINARIES

We first provide a brief description of PLSA as background and introduce the term *Simplex Decomposition*. Let the data be given by the $W \times D$ matrix $\mathbf{C}$, with the (wd)-th element given by c_{wd}. In this section, the model is explained in the context of word-counts data for the purposes of exposition.

2.1. Probabilistic Latent Semantic Analysis

PLSA is a statistical model that characterizes a corpus of text documents by extracting a set of *latent factors* or *topics* that the corpus is comprised of. Data is modeled by parametrizing it in terms of the following multinomial distributions - the probability of word w appearing in document d, $P(w|d)$; the probability of word w appearing in topic t, $P(w|t)$; and the probability of topic t appearing in document d, $P(t|d)$. Mathematically, the model can now be written as

$$P(w,d) = P(d)P(w|d) = P(d)\sum_t P(w|t)P(t|d). \quad (1)$$

Consider the column $\mathbf{c}_d$ belonging to the d-th document from the data set $\mathbf{C}$. The normalized version of this vector $\bar{\mathbf{c}}_d$ (obtained by scaling the entries to sum to 1.0), which we shall refer to as a *data distribution*, is a multinomial distribution underlying the document. The model approximates this data distribution by $P(w|d)$,

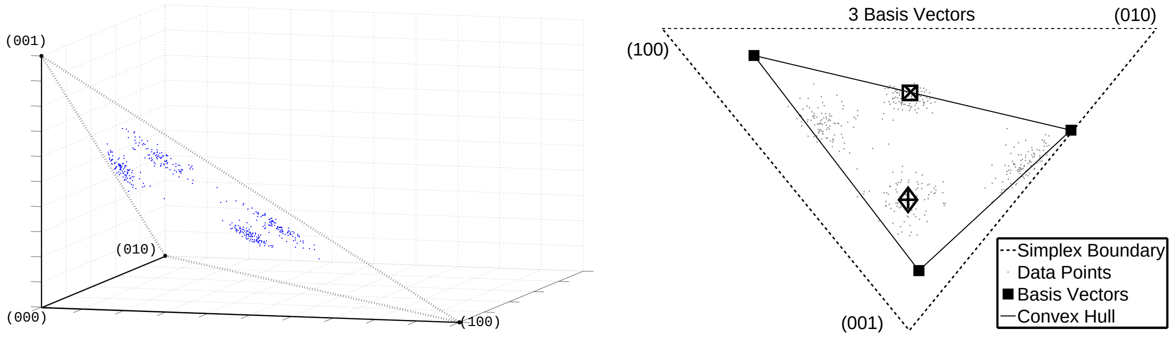

Fig. 1. Illustration of PLSA. The left panel shows an artificial data set of 400 3-dimensional data distributions. The data distributions, shown as blue points lie on the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$, shown as a dotted-triangle. The right panel shows the Standard 2-Simplex as a 2-dimensional plane along with a set of 3 topic distributions (basis vectors) extracted from the data set using PLSA. The model approximates data distributions as points lying within the convex hull formed by the topic distributions. Also shown are two data points (marked by + and $\times$) and their respective approximations (respectively shown as $\diamondsuit$ and $\Box$).

which in turn is expressed as a linear combination of *topic distributions* $P(w|t)$. The topic distributions can be thought of as a set of basis vectors that combine in different proportions (given by $P(t|d)$) to form the data distributions.

Given the matrix $\mathbf{C}$, parameters can estimated through iterations of the following equations derived using the EM algorithm,

$$P(t|w,d) = \frac{P(t|d)P(w|t)}{\sum_t P(t|d)P(w|t)}, \quad \text{and}$$

$$P(w|t) = \frac{\sum_d c_{wd} P(t|w,d)}{\sum_w \sum_d c_{wd} P(t|w,d)}, P(t|d) = \frac{\sum_w c_{wd} P(t|w,d)}{\sum_w c_{wd}} \tag{2}$$

The EM algorithm guarantees that the above updates converge to a local optimum.

We briefly point out here that the PLSA model of equation (1) can be written as a matrix factorization as follows:

$$\bar{\mathbf{C}}_{W\times D} \approx \mathbf{W}_{W\times K}\mathbf{H}_{K\times D} = \mathbf{P}_{W\times D}, \tag{3}$$

where $\bar{\mathbf{C}}$ is the matrix of data distributions (normalized data set), $\mathbf{W}$ is the topic matrix of entries $P(w|t)$ with column $\mathbf{w}_t$ corresponding to the t-th topic, $\mathbf{H}$ is the mixture-weights matrix of entries $P(t|d)$ with column $\mathbf{h}_d$ corresponding to the d-th document and $\mathbf{P}$ is the model approximation matrix of entries $P(w|d)$ with column $\mathbf{p}_d$ corresponding to the d-th document. It has been pointed out before [4, 5] that this decomposition is equivalent to Non-negative Matrix Factorization.

2.2. PLSA as Simplex Decomposition

The PLSA model can be visualized geometrically as illustrated in Figure 1. The normalized datapoints $\bar{\mathbf{c}}_d$, the model approximations $\mathbf{p}_d$, and the topic distributions $\mathbf{w}_t$, all being W-dimensional multinomial distributions, can be viewed as points in a $(W-1)$ simplex. The model expresses $\mathbf{p}_d$ as points within the convex hull formed by topic distributions $\mathbf{w}_t$. The aim of the model is to determine topics $\mathbf{w}_t$ such that the model $\mathbf{p}_d$ for any normalized data point $\bar{\mathbf{c}}_d$ approximates it closely. Since the model $\mathbf{p}_d$ is constrained to lie within the simplex defined by $\mathbf{w}_t$, it can model $\bar{\mathbf{c}}_d$ accurately only if the latter also lies within the same hull. Any data distribution $\bar{\mathbf{c}}_i$ that lies outside the hull defined by $\mathbf{w}_t$ is modeled with error. Thus, the objective of the model is to find topics (or basis vectors) such that they form the corners of a simplex enclosing the normalized data points. We christen any model that decomposes a dataset into a set of basis vectors that form a simplex around the datapoints as a *Simplex Decomposition*. As we have seen, PLSA is one such example of a Simplex Decomposition.

One of the advantages of Simplex Decompositions is that it allows for principled extensions. One can take PLSA as an example where several extensions have been proposed by imposing additional structure on the mixture weights. These extensions such as LDA (with a Dirichlet prior on mixture weights), Sparse-PLSA (entropic prior on mixture weights, [6]) and Correlated Topic Model (with a log-normal prior on mixture weights) also are examples of Simplex Decompositions since they too extract basis vectors that form corners of a simplex.

All these Simplex Decompositions we have mentioned so far operate only on non-negative data. The next section describes the main contribution of this paper where we generalize this idea to be applicable to real-valued datasets with ± entries.

3. SIMPLEX DECOMPOSITION FOR ARBITRARY DATASETS

Consider a new problem where we are given a $W \times D$ data matrix $\mathbf{B}$. Unlike $\mathbf{C}$, the entries of $\mathbf{B}$ are no longer constrained to be non-negative. The problem is to find a Simplex Decomposition similar to equation (3). That means we need a set of "basis vectors" that combine additively without any cross-cancellations (*i.e.* with only non-negative co-efficients) to approximate the given dataset and form the corners of a simplex surrounding the datapoints (*i.e.* the co-efficients sum to 1). In other words, we desire a decomposition of the form

$$\mathbf{B} \approx \boldsymbol{\mathcal{W}}_{W\times K}\boldsymbol{\mathcal{H}}_{K\times D} = \boldsymbol{\mathcal{P}}_{W\times D} \tag{4}$$

where K is the desired dimensionality of the decomposition, $\boldsymbol{\mathcal{W}}$ is the matrix of basis vectors $[\boldsymbol{\omega}_1 \ldots \boldsymbol{\omega}_K]$, $\boldsymbol{\mathcal{H}}$ is the matrix of mixture-weights $[\hbar_1 \ldots \hbar_D]$, and $\boldsymbol{\mathcal{P}}$ is the matrix of approximations $[\boldsymbol{\rho}_1 \ldots \boldsymbol{\rho}_D]$. Note that entries of the basis vectors (entries of matrix $\boldsymbol{\mathcal{W}}$) are not constrained to be non-negative[1].

[1] We use PLSA as the example topic model to illustrate the method but the proposed techniques are applicable to any topic model that expresses

The basic idea of the method is as follows. We transform the data vectors in $\mathbf{B}$ such that the transformed data vectors are multinomial distributions (all entries are non-negative and sum to 1). PLSA can then be applied on this transformed dataset to obtain basis vectors and mixture weight matrices. The basis vectors thus obtained in the transformed space are transformed back into the original data space. If we make sure that the transformation is linear, the mixture weights obtained from the PLSA step will also serve as the mixture weights for the decomposition in the original data space. We call this new method *Real*-PLSA (or more generally *Real*-`model` where `model` denotes any other topic model).

3.1. Naive Approach

The simplest way to transform a dataset with ± entries to one with only positive entries is a simple translation. By adding a number greater than the magnitude of the minimum entry in the dataset, all entries of the dataset can be turned positive.

However, this straight-forward approach will be insufficient. To understand, note that PLSA (and other topic models) does not model the given non-negative dataset directly but models the underlying distributions. In other words, PLSA models the *normalized* dataset. This implies that the geometry of the normalized dataset differs based on the magnitude of the offset applied during the translation.

Consider an example of three two dimensional points - $A(-2.5, -0.5)$, $B(-1.5, 1.5)$ and $C(-2, 0.5)$. The relative distances between the points are

$$|AB| : |BC| : |CA| :: 2 : 1 : 1. \tag{5}$$

Now, consider adding a positive number 3.5 to all the entries to make the dataset positive. We obtain points $A(1, 3)$, $B(2, 5)$ and $C(1.5, 4)$. What PLSA tries to model, however, is the normalized version of these points, given by $A'(0.25, 0.75)$, B'(0.2857, 0.7143) and $C'(0.2727, 0.7273)$. The relative distances for these normalized points are given by

$$|A'B'| : |B'C'| : |C'A'| :: 2.75 : 1 : 1.75.$$

Adding 3.5 turned all entries positive but it skewed the relative geometry of the normalized points. Adding a different constant instead of 3.5 will skew the geometry differently. For example, adding 4.5 to the points gives us $A(2, 4)$, $B(3, 6)$ and $C(2.5, 5)$, all of whom correspond to the point $(1/3, 2/3)$ when normalized.

This example clearly demonstrates the skew introduced by the above approach. We need a transformation that also preserves the relative geometry between all the points. This can be accomplished by transforming the data to not only have positive entries but to make sure they are also multinomial distributions, as we explain in the following subsection.

3.2. Real Topic Models

The transformation we propose is based on a very simple intuition. Given a real-valued dataset of dimensionality W with ± entries, it is always possible to express it as a $(W + 1)$-dimensional dataset where all the points lie on an W-dimensional hyperplane. However, we need a method to identify the hyperplane we desire. We know that for PLSA to be applicable on the new transformed dataset, we need the points to lie on a hyperplane such that all entries are positive and sum to 1[2]. The latter constraint that all entries sum to 1 actually *defines* the W-dimensional hyperplane we need. For example, if we are given a 2-dimensional data set, we need the hyperplane $x_1 + x_2 + x_3 = 1$ in 3D space where variables x_1, x_2 and x_3 indicate the co-ordinates of the 3D space. A simple transformation onto this hyperplane is not enough - the first constraint implies that we need the points to lie in the portion of the plane that is in the positive orthant of the $(W + 1)$-dimensional space. This region of the hyperplane, marked by points unit-distance away from the origin on the positive side of each of the $(W + 1)$ axes, forms the standard W-simplex. In the 3D case, this simplex is given by the triangle formed by points (001), (010) and (100).

To accomplish the first step, we need a set of orthonormal vectors that define the desired W-dimensional hyperplane in $(W+1)$-dimensions. There are infinitely many different sets of orthonormal basis vectors that can be used and as many different ways to find one such set. The first contribution of this paper is a method to obtain one such orthonormal basis. We use a simple recursive method and the details are explained in Appendix A.

Let $\mathbf{T}$ denote the $(W+1) \times W$ orthonormal matrix where the columns represent individual basis vectors that define the desired hyperplane. For example, in the case of 2-dimensional data, the two unit-vectors defining the plane are given by $k1 \times [-1, 1, 0]$ and $k2 \times [1, 1, -2]$ where $k1 = 1/\sqrt{2}$ and $k2 = 1/\sqrt{6}$ are appropriate normalizing factors.

The co-ordinates of the original dataset in the $(W + 1)$ dimensional space can now be computed as the columns of matrix

$$\hat{\mathbf{B}} = (\mathbf{B}^T\mathbf{T}^T)^T = \mathbf{TB}. \tag{6}$$

The new transformed $(W + 1)$-dimensional data points in $\hat{\mathbf{B}}$ lie in a space parallel to the desired hyperplane.

The second step is to make sure that all transformed data points are within the standard W-simplex, *i.e.* all entries are positive. We check the matrix $\hat{\mathbf{B}}$ for negative entries and subtract the least negative entry from the entire matrix. This is equivalent to translating all the datapoints in a direction orthogonal to the simplex until all the co-ordinates are non-negative. Because of the translation, the entries of each datapoint sum to constant, say Q. We rescale the data by Q to transform them to lie within the standard W-simplex. Let the data after this stage of transformation be denoted by the $(W + 1) \times D$ matrix $\bar{\mathbf{B}}$.

Consider the example points from the previous section $A(-2.5, -0.5)$, $B(-1.5, 1.5)$ and $C(-2, 0.5)$, whose relative distances were given by equation (5). Transforming these points as mentioned above, we obtain three-dimensional points A'(0, 0.5977, 0.4023), $B'(0.2576, 0.6161, 0.1263)$ and $C'(0.1288, 0.6069, 0.2643)$. It is easy to verify that

$$|A'B'| : |B'C'| : |C'A'| :: 2 : 1 : 1,$$

thus the transformation preserves the local geometry of the points.

The dataset $\bar{\mathbf{B}}$ is now amenable for PLSA as the entries of each column are positive and sum to 1. Performing PLSA, we obtain

$$\bar{\mathbf{B}} \approx \bar{\boldsymbol{\mathcal{W}}}_{(W+1)\times K}\boldsymbol{\mathcal{H}}_{K\times D} = \bar{\boldsymbol{\mathcal{P}}}_{(W+1)\times D},$$

document distributions as mixtures of topics similar to PLSA.

[2] Technically, data vectors need not sum to 1 for PLSA to be applicable. However, what PLSA models is in fact normalized data vectors. Having this constraint explicitly simplifies the choice of our data transformation.

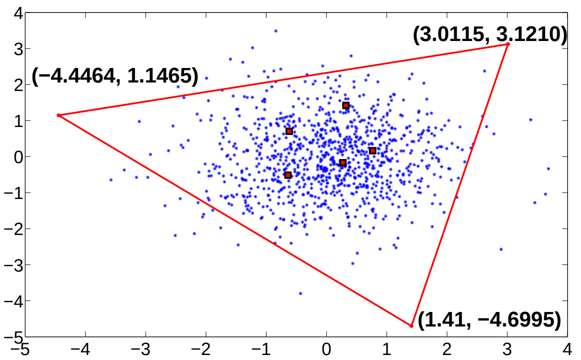

Fig. 2. Illustration of *Real*-PLSA. Blue dots show 1000 two-dimensional datapoints generated by a Gaussian distribution. Corners of the red triangle correspond to the basis vectors extracted. Also shown are five randomly chosen points from the dataset illustrated as red stars and their corresponding model approximations shown by black squares (see text). 37 data-points lie outside the red triangle and are thus modeled with error.

where $\bar{\mathcal{W}}$ and $\bar{\mathcal{P}}$ are matrices of basis vectors and approximations expressed in the $(W+1)$-space, and $\mathcal{H}$ is the matrix of mixture weights.

We now subject $\bar{\mathcal{W}}$ to the same transformations that converted $\mathbf{B}$ to $\bar{\mathbf{B}}$ but in the reverse order. We first undo from $\bar{\mathcal{W}}$ the scaling and translation that matrix $\hat{\mathbf{B}}$ was applied with to obtain $\bar{\mathbf{B}}$. Let us denote the basis vector matrix at this stage as $\hat{\mathcal{W}}$. The final step is to transform these vectors into the original W-dimensional space, which is accomplished by[3]

$$\mathcal{W} = (\hat{\mathcal{W}}^T \mathbf{T})^T. \tag{7}$$

Since all the transformations involved were linear, it also implies that

$$\mathbf{B} \approx \mathcal{W}_{W\times K}\mathcal{H}_{K\times D}.$$

The algorithm is summarized in Appendix B in matlab code which is also available at `http://cns.bu.edu/~mvss/upub/`.

3.3. Illustrations on Synthetic Datasets

3.3.1. Example 1

Let us take an example of 1000 points in 2-dimensions generated by a Gaussian distribution to understand the procedure. Shown below are five points randomly chosen from this data that we will use to illustrate every step of the process:

$$\begin{bmatrix} -0.6179 & -0.6371 & 0.3201 & 0.2714 & 0.7628 \\ 0.7010 & -0.5129 & 1.4165 & -0.1687 & 0.1634 \end{bmatrix}$$

The first step is to transform the dataset by projecting onto the vectors of $\mathbf{T}$ as given by equation (6). We obtain the transformed datapoints as

$$\begin{bmatrix} -0.1507 & -0.6599 & 0.8046 & 0.1230 & 0.6061 \\ 0.7231 & 0.2411 & 0.3519 & -0.2607 & -0.4727 \\ -0.5723 & 0.4187 & -1.1565 & 0.1377 & -0.1334 \end{bmatrix}$$

The minimum entry in the new transformed dataset turns out to be -3.1004. Subtracting that from all the entries and re-scaling them by the sum of the columns, we obtain the points as

$$\begin{bmatrix} 0.3171 & 0.2624 & 0.4198 & 0.3466 & 0.3985 \\ 0.4111 & 0.3593 & 0.3712 & 0.3053 & 0.2825 \\ 0.2718 & 0.3784 & 0.2090 & 0.3481 & 0.3190 \end{bmatrix}.$$

Running PLSA on this transformed dataset, we obtain the basis vectors as

$$\begin{bmatrix} 0.6993 & 0.2343 & 0.0456 \\ 0.2414 & 0.0199 & 0.7217 \\ 0.0594 & 0.7459 & 0.2327 \end{bmatrix}$$

which when transformed back to the original 2-dimensional space can be written as

$$\begin{bmatrix} 3.0115 & 1.4100 & -4.4464 \\ 3.1210 & -4.6995 & 1.1465 \end{bmatrix},$$

and the mixture weights corresponding to the five points are also obtained as shown below.

$$\begin{bmatrix} 0.3586 & 0.2272 & 0.5345 & 0.3605 & 0.4481 \\ 0.1968 & 0.3607 & 0.1337 & 0.3467 & 0.3194 \\ 0.4445 & 0.4121 & 0.3318 & 0.2928 & 0.2324 \end{bmatrix}.$$

Figure 2 illustrates the results.

3.3.2. Example 2

We now take another example where we created a dataset by taking a set of basis vectors and combining them using randomly generated mixture weights. The goal in this case is to test whether the proposed algorithm can extract the original basis vectors when applied on the synthesized dataset.

The dataset was synthesized as follows. Three basis vectors containing both positive and negative entries were first generated and combined with 50 randomly generated mixture weights to synthesize the dataset. The mixture weight entries corresponding to each data point were all positive and totaled to 1. The matlab commands used to generate the data were as follows:

```
B = interp2( randn( 10, 3), (1:3)',...
            (1:.1:10), 'spline');
H = rand( 3, 50).^3;
H = H ./ repmat((sum(H)), 3, 1);
```

where `B` corresponds to the matrix of basis vectors, `H` is the matrix of mixture weights, and the synthesized dataset is given by the matrix product `BH`.

The orginal basis vectors, mixture weights and a subset of the data points are shown in Figure 3. Also shown are the bases extracted by applying the *Real*-PLSA algorithm on this dataset. We point out that the algorithm successfully extracts the original bases with which the data was synthesized. This example illustrates the ability of the algorithm to uncover the original structure in a dataset.

The application illustrated by this example is similar to Independent Components Analysis (ICA). Figure 3 also shows the bases obtained by ICA (using the FastICA[4] algorithm) for comparison and notice that the *Real*-PLSA bases are better than the

[3] Since $\mathbf{T}$ is orthonormal, $\mathbf{T}$ in equation (7) inverses the transformation done by $\mathbf{T}^T$ in equation (6). If $\mathbf{T}$ is not normalized and is just an orthogonal matrix, one has to use $pinv(\mathbf{T}^T)$ instead of $\mathbf{T}$ in equation (7), where $pinv(.)$ denotes the pseudoinverse.

[4] http://www.cis.hut.fi/projects/ica/fastica/

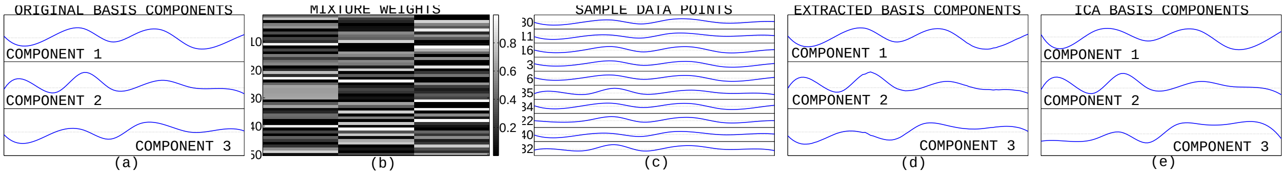

Fig. 3. Illustration of *Real*-PLSA on a synthesized dataset. 50 data points were synthesized by combining the three basis vectors shown in panel (a) with 50 randomly generated triplets. All the triplets sum to 1 and are shown in panel (b). A subset of 10 randomly chosen data points thus constructed are shown in panel (c). *Real*-PLSA was applied to extract three basis vectors from this synthesized dataset, and the resulting bases extracted are shown in panel (d). For comparison, bases obtained from Independent Components Analysis (ICA) are shown in panel (e). A simple root-mean-squared error comparison between the extracted and original components show that *Real*-PLSA is better (RMS errors 0.06, 0.21 & 0.42 for the three components respectively) compared to ICA (RMS errors 0.29, 0.34 & 0.7 respectively).

ICA bases. Unlike ICA algorithms, *Real*-PLSA decomposition is a Simplex Decomposition where the mixture weights are all nonnegative and sum to 1. Since this constraint is built into the decomposition, the constraint was also incorporated while synthesizing the dataset so that the algorithm's ability to uncover the original bases components could be tested. Nevertheless, relationships between the proposed algorithm and ICA is a ripe area for research that we leave for future work.

4. DISCUSSION

In this section, we briefly discuss the complexity issues of the proposed method and present some related work. We first point out that Real Topic Models we have presented, even though inspired by applications on ± data, can also be applicable to non-negative data in certain situations. As we have pointed out, PLSA[5] and its extensions model not the data directly but the data distributions. In some situations where the "energies" of data points are also important, modeling data distributions is not desirable and one can model datapoints directly by resorting to the proposed approach.

4.1. Complexity

The core of the proposed technique is the original topic model itself. The data is preprocessed first to make it ready for the topic model and once the basis vectors are obtained from the topic model, they are processed back to the original data space. The most intensive step in the preprocessing of data involves a matrix multiplication. Given this fact and the fact that most topic models employ iterative algorithms to estimate parameters, the topic model core computations act as the complexity bottleneck. Thus, the complexity of the proposed approach depends on the complexity of the algorithms used for the topic model computations.

4.2. Related Work

The author is not aware of any work in the topic modeling community to extend their methodologies for data with negative and positive entries. It is understandable as the main motivation there is to analyze text corpora where one does not encounter such datasets. However, there has been some work to extend the technique of Nonnegative Matrix Factorization to generalized datasets. A technique called Semi-Nonnegative Matrix Factorization was introduced by [7] where the goals were similar to ours. In particular, the method generates a decomposition of the form

$$\mathbf{X}_{\pm} \approx \mathbf{F}_{\pm}\mathbf{G}_{+}^{T}$$

where the subscripts of the matrices indicate the signs of entries allowed in the matrices. Even though the constraints imposed in the decomposition are similar to what we propose, the similarities end there. Since there are no additional constraints imposed on the matrix $\mathbf{G}^T$, the method is not a Simplex Decomposition and the extracted bases cannot be viewed as corners of a convex-hull surrounding the dataset. In fact, it is argued in the paper, despite unsatisfactory results, that the obtained basis vectors are analogues of centroids obtained from a clustering algorithm on the data. In addition to this advantage of interpretability, the main advantage of the technique presented in this paper over this method is the generality of the approach. The approach of Real Topic Models allows one to model data using any extension of PLSA whereas in Semi-Nonnegative Matrix Factoization, there is no principled way to impose priors such as dirichlet or log-normal distributions. And the modular nature of our approach ensures that any future algorithmic advances in estimation can be incorporated right away.

There has been a lot of work in the computational geometry field in tackling the problem of constructing convex-hulls for arbitrary dimensional data. For a long time, solutions were known only for even-dimensional data (and the 3-dimensional case) until [8] finally proposed a general solution. Once a convex-hull is found, theoretically every data-point can be represented as a mixture of points on the corners of the hull though the algorithms do not explicitly compute the mixture weights. In computational geometry, the focus is on finding a *full* convex hull and obtaining exact solutions. In real topic models, the number of basis vectors one desires is often far lesser than the dimensionality of the dataset. Thus, the basis vectors can be thought of representing a convex-hull-like structure in lower dimensions. And topic models do not produce exact results as iterative algorithms such as EM or variational methods are often employed to arrive at the solutions. In spite of these differences, it will be instructive to compare the two approaches and we leave that for future work.

5. CONCLUSIONS

In this paper, we introduced a new method that enables one to apply topic models such as PLSA, LDA and other extensions of PLSA on data that has both negative and positive entries. More

[5] There is another factorization model also sometimes referred to as PLSA given by $P(w,d) = \sum_t P(t)P(w|t)P(d|t)$. This new model also can be applied on ± data using the proposed method to obtain ± bases. We skip details due to lack of space.

specifically, we introduced the idea of a *Simplex Decomposition* - of which PLSA and extensions are examples - and showed how it can be generalized to data with ± entries. A Simplex Decomposition decomposes a dataset into basis vectors that form the corners of a simplex surrounding the dataset. We showed this geometry in the case of PLSA. We then described how any arbitrary dataset can be linearly transformed into the next higher dimension where they are represented as data distributions. This allows the application of topic models on the transformed dataset. We presented an algorithm to compute the transformation and illustrated the method by example applications on synthetic datasets. We also discussed the complexity issues of the proposed algorithm and reviewed related work. There are several potential applications of this work for tasks such as feature extraction, clustering and classification among others. The work brings the power of statistical topic models to datasets without the need to have non-negativity constraints. Among other directions, a promising area of future work would be to compare and contrast the performance of these topic models with standard machine learning algorithms.

Acknowledgements: I thank Paris Smaragdis at Adobe and Bhiksha Raj at CMU for invaluable comments and feedback.

Appendices

A. CONSTRUCTION OF T

In this appendix, we propose a method to generate a set of W orthonormal $(W+1)$-dimensional vectors that span the standard W-simplex. We first find a set of orthogonal vectors which are normalized later. Notice that since each vector of the matrix lies parallel to the simplex, the sum of all entries in the vector should sum to zero. The entries of every point within the simplex sums to the same constant and a vector, being the difference of two points, will have entries that sum to zero. Given this constraint, an orthogonal set of vectors can be found in an inductive fashion based on the two basic observations. Let $\mathbf{T}_W$ denote a $W \times (W-1)$ matrix of $(W-1)$ orthogonal vectors. Let $\vec{\mathbf{1}}_W$ and $\vec{\mathbf{0}}_W$ denote W-vectors where all the entries are 1's and 0's respectively. Similarly, let $\mathbf{1}_{a\times b}$ and $\mathbf{0}_{a\times b}$ denote $a \times b$ matrices of all 1's and 0's respectively. It can be easily shown that the matrix $\mathbf{T}_{(W+1)}$ given by

$$\begin{bmatrix} \mathbf{T}_W & \vec{\mathbf{1}}_W \\ \vec{\mathbf{0}}^T_{(W-1)} & -W \end{bmatrix} \quad \text{if } W \text{ is even, and}$$

$$\begin{bmatrix} \mathbf{T}_{(W+1)/2} & \mathbf{0}_{(W+1)/2\times(W-1)/2} & \vec{\mathbf{1}}_{(W+1)/2} \\ \mathbf{0}_{(W+1)/2\times(W-1)/2} & \mathbf{T}_{(W+1)/2} & -\vec{\mathbf{1}}_{(W+1)/2} \end{bmatrix},$$

if W is odd, is orthogonal. $\mathbf{T}_{(W+1)}$ is then normalized to obtain an orthonormal matrix.

Given these rules and the fact that $\mathbf{T}_1$ is an empty matrix, one can easily build an algorithm to find the matrix $\mathbf{T}_W$. Pseudocode for the algorithm is shown below.

```
% m - dimensionality of data
% Find (m+1)-by-m orthonormal matrix tc.
idxs = [m+1]; i = m+1; tc = [];
while (i˜=1)
   idxs = [idxs floor(i/2)];
   i = floor(i/2);
end
for i=idxs(end-1:-1:1)
  tcs2 = size(tc, 2);
  tn = [tc*[eye(tcs2) zeros(tcs2, tcs2)];...
        tc*[zeros(tcs2, tcs2) eye(tcs2)]];
  tn = [tn [ones(floor(i/2), 1);...
            -ones(floor(i/2), 1)]];
  if rem(i, 2)
    tn = [tn ones(i-1, 1)];
    tn = [tn; [zeros(1, i-2) -(i-1)]];
  end
  tc = tn;
end
tc = tc./repmat(sqrt(sum(tc.ˆ2)), m+1, 1);
```

B. REAL TOPIC MODELS: ALGORITHM

This appendix provides matlab code for the entire algorithm.

```
% Inputs: m-by-n data matrix v, Number of
%         desired components r.
% Outputs: m-by-r basis vector matrix w
%          r-by-n mixture-weight matrix h
% Form the orthonormal transformation
% matrix tc (see Appendix A)

% transform the data into (n+1) dimensions
nv = (v’*tc’)’;
mnv = min(nv(:)); nv = nv-mnv;
msc = sum(nv(:, 1)); nv = nv ./ msc;

% Run Topic Model
[w, h] = topic_model(nv, r, param);

% Transform w back to orig dimensions
w = w*msc; w = w+mnv;
w = (w’*tc)’;
```

C. REFERENCES

[1] T Hofmann, “Unsupervised Learning by Probabilistic Latent Semantic Analysis,” *Machine Learning*, vol. 42, 2001.

[2] DM Blei, AY Ng, and MI Jordan, “Latent Dirichlet Allocation,” *Jrnl of Machine Learning Res.*, vol. 3, 2003.

[3] DM Blei and JD Lafferty, “Correlated Topic Models,” in *NIPS*, 2006.

[4] E Gaussier and C Goutte, “Relation between PLSA and NMF and Implications,” in *Proc. ACM SIGIR Conf. on Research and Dev. in Information Retrieval*, 2005, pp. 601–602.

[5] M Shashanka, B Raj, and P Smaragdis, “Probabilistic latent variable models as non-negative factorizations,” *Computational Intelligence and Neuroscience*, May 2008.

[6] M Shashanka, B Raj, and P Smaragdis, “Sparse Overcomplete Latent Variable Decomposition of Counts Data,” in *NIPS*, 2007.

[7] Chris Ding, Tao Li, and Michael Jordan, “Convex and semi-nonnegative matrix factorizations,” Tech. Rep. 60428, Lawrence Berkely National Laboratory, 2006.

[8] Bernard Chazelle, “An optimal convex hull algorithm in any fixed dimension,” *Discrete & Comp. Geometry*, vol. 10, 1993.