A Purely Geometric Approach to Non-Negative Matrix Factorization

Christian Bauckhage

B-IT, University of Bonn, Bonn, Germany
Fraunhofer IAIS, Sankt Augustin, Germany
http://mmprec.iais.fraunhofer.de/bauckhage.html

Abstract. We analyze the geometry behind the problem of non-negative matrix factorization (NMF) and devise yet another NMF algorithm. In contrast to the vast majority of algorithms discussed in the literature, our approach does not involve any form of constrained gradient descent or alternating least squares procedures but is of purely geometric nature. In other words, it does not require advanced mathematical software for constrained optimization but solely relies on geometric operations such as scaling, projections, or volume computations.

Keywords: latent factor models, data analysis.

1 Introduction

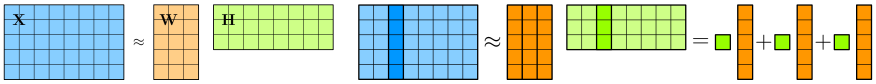
Non-negative matrix factorization (NMF) has become a popular tool of the trade in areas such as data mining, pattern recognition, or information retrieval. Ever since Paatero and Tapper [14] and later Lee and Seung [11] published seminal papers on NMF and its possible applications, the topic has attracted considerable research that produced a vast literature. Related work can be distinguished into two main categories: either reports on practical applications in a wide range of disciplines or theoretical derivations of efficient algorithms for NMF.

The work reported here belongs to the latter category. However, while our technique scales to very large data sets, our focus is not primarily on efficiency. Rather, our main goal is to expose a new point of view on NMF and to show that it can be approached from an angle that, to the best of our knowledge, has not been widely considered yet.

In order for this paper to be accessible to a wide audience, we first review the NMF problem, its practical appeal, established algorithms for its computation, and known facts about its complexity. Readers familiar with matrix factorization for data analysis might want to skip this introductory exposition.

Then, we discuss NMF from a geometric point of view and devise an NMF algorithm that does not involve gradient descent or alternating least squares

In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at http://ceur-ws.org



(a) data- and factor matrices (b) interpretation of matrix factorization

Fig. 1: Visualization of the idea of matrix factorization and its interpretation as representing data vectors in terms of linear combinations of a few latent vectors.

schemes. Rather, our approach is based on strikingly simple geometric properties that were already noted by Donoho and Stodden [7] and Chu and Lin [4] but, again to our best knowledge, have not yet been fully exploited to design NMF algorithms. In short, we present an approach towards computing NMF that does not explicitly solve constrained optimization problems but only relies on rather simple operations.

The three major benefits we see in this are: (a) our approach allows users to compute NMF even if they do not have access to specialized software for numerical optimization; (b) it allows for parallelization and therefore naturally scales to BIG DATA settings; (c) last but not least our approach hardly requires prior knowledge as to optimization theory and convex analysis and therefore provides an alternative, possibly more intuitive avenue towards teaching these materials to students.

2 Non-Negative Matrix Factorization

Applications of NMF naturally arise whenever we are dealing with the analysis of data that reflect counts, ranks, or physical measurements such as weights, heights, or circumferences which are non-negative by definition. In situations like these, the basic approach is as follows: Assume a set $\{\boldsymbol{x}_j\}_{j=1}^n$ of n non-negative data vectors $\boldsymbol{x}_j \in \mathbb{R}^m$ and gather them in an $m \times n$ data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$.

Given such a non-negative data matrix, we write $\boldsymbol{X} \succeq \boldsymbol{0}$ to express that its entries $x_{ij} \geq 0$ for all i and j. The problem of computing a non-negative factorization of $\boldsymbol{X}$ then consists of two basic tasks:

1. Fix an integer $k \ll \operatorname{rank}(\boldsymbol{X}) \leq \min(m, n)$.
2. Determine two non-negative factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ where $\boldsymbol{W}$ is of size $m \times k$, $\boldsymbol{H}$ is of size $k \times n$, and their product approximates $\boldsymbol{X}$. In other words, determine two non-negative, rank-reduced matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$. Mathematically, this can be cast as a constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \; E(k) = & \left\| \boldsymbol{X} - \boldsymbol{W}\boldsymbol{H} \right\|^2 \\ \text{subject to} \quad & \boldsymbol{W} \succeq \boldsymbol{0} \\ & \boldsymbol{H} \succeq \boldsymbol{0} \end{aligned} \tag{1}$$

where $\|\cdot\|$ denotes the matrix Frobenius norm. Note that instead of minimizing a matrix norm to determine suitable factor matrices, we could also attempt to minimize (more) general divergence measures $D(\boldsymbol{X}||\boldsymbol{WH})$. Yet, w.r.t. actual computations this would not make much of a difference so that we confine our discussion to the more traditional norm-based approaches.

Now, assume, for the time being, that $\boldsymbol{W}$ and $\boldsymbol{H}$ have been computed already. Once they are available, it is easy to see that each column $\boldsymbol{x}_j$ of $\boldsymbol{X}$ can be reconstructed as

$$\boldsymbol{x}_j \approx \hat{\boldsymbol{x}}_j = \boldsymbol{W}\boldsymbol{h}_j = \sum_{i=1}^{k} \boldsymbol{w}_i h_{ij} \tag{2}$$

where $\boldsymbol{h}_j$ denotes column j of $\boldsymbol{H}$ and $\boldsymbol{w}_i$ refers to the ith column of $\boldsymbol{W}$. Next, we briefly point out general benefits and applications of this representation of the given data.

2.1 General Use and Applications

Looking at (2), the following properties and corresponding applications of data matrix factorization quickly become apparent:

Latent component detection: Each data vector $\boldsymbol{x}_j$ is approximated in terms of a linear combination of the k column vectors $\boldsymbol{w}_i$ of matrix $\boldsymbol{W}$. Thus, in a slight abuse of terminology, $\boldsymbol{W}$ is typically referred to as the matrix of "basis vectors". Since each $\boldsymbol{w}_i$ is an m-dimensional vector, any linear combination of the $\boldsymbol{w}_i$ produces another m-dimensional vector. Yet, since the number k of basis vectors in $\boldsymbol{W}$ is less than the dimension m of the embedding space, we see that the reconstructed data vectors $\hat{\boldsymbol{x}}_j$ reside in a k-dimensional subspace spanned by the $\boldsymbol{w}_i$. Hence, solving (1) for $\boldsymbol{W}$ provides k latent factors $\boldsymbol{w}_i$ each of which characterizes a different distinct aspect or tendency within the given data.

Dimensionality reduction: There is a one-to-one correspondence between the data vectors $\boldsymbol{x}_j$ in $\boldsymbol{X}$ and the columns $\boldsymbol{h}_j$ of $\boldsymbol{H}$ and we note that the entries h_{ij} of vector $\boldsymbol{h}_j$ assume the role of coefficients in (2). Accordingly, the factor matrix $\boldsymbol{H}$ is typically referred to as the coefficient matrix. We also note that while $\boldsymbol{x}_j$ is an m-dimensional vector, the corresponding coefficient vector $\boldsymbol{h}_j$ is only k-dimensional. In this sense, NMF implicitly maps m-dimensional data to k-dimensional representations.

Data compression: Storage requirements for the original data matrix $\boldsymbol{X}$ are of the order of $O(mn)$. For the approximation $\boldsymbol{X} \approx \boldsymbol{WH}$, however, we would only have to store an $m \times k$ and a $k \times n$ matrix which would need space of the order of $O(k(m+n))$. Since typically $k \ll mn/(m+n)$ this allows for considerable savings.

All these practical benefits also apply to related methods such as the singular value decomposition (SVD) or independent component analysis to name but a few. In this sense, NMF is not at all special. However, while methods such as the SVD are well appreciated for their statistical guarantees as to the quality of the resulting low-rank data representations, they are not necessarily faithful to the nature of the data. In other words, basis vectors resulting from other methods are usually not non-negative and therefore will explain non-negative data in terms of latent factors that may not have physical counterparts. It is for reasons like these that NMF has become popular.

2.2 General Properties and Characteristics

Looking at (1), we recognize a problem that is convex in either $\boldsymbol{W}$ *or* $\boldsymbol{H}$ but not in $\boldsymbol{W}$ *and* $\boldsymbol{H}$ jointly. In other words, NMF suffers from the fact that the objective function $E(k)$ usually has numerous local minima. Although a unique global minimum provably exists [20], there are no algorithms known today that were guaranteed to find it within reasonable time.

Indeed, (1) is an instance of a constrained Euclidean sum-of-squares problem and thus NP hard [1, 21]. Consequently, known NMF algorithms typically approach the problem using iterative procedures. Usually, both factor matrices are randomly initialized to non-negative values and then refined by means of alternating least squares or gradient descent schemes.

The former approach goes back to [14] and works like this: first, fixate $\boldsymbol{W}$ and solve (1) for $\boldsymbol{H}$ using non-negative least squares solvers. Then, given the updated coefficient matrix, solve (1) for $\boldsymbol{W}$ and repeat both steps until convergence.

The latter idea was first considered in [11] and makes use of the fact that

$$E(k) = \left\| \boldsymbol{X} - \boldsymbol{W}\boldsymbol{H} \right\|^2 = \operatorname{tr}\left[\boldsymbol{X}^T\boldsymbol{X} - 2\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} \right] \tag{3}$$

so that

$$\frac{\partial E}{\partial \boldsymbol{W}} = 2\left[\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{X}\boldsymbol{H}^T \right] \quad \text{and} \quad \frac{\partial E}{\partial \boldsymbol{H}} = 2\left[\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} - \boldsymbol{W}^T\boldsymbol{X} \right]. \tag{4}$$

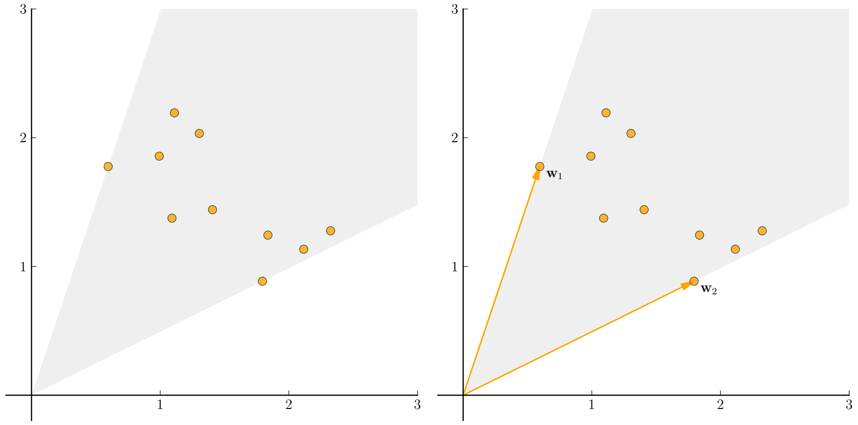
Updates for both factor matrices can thus be computed in another alternating fashion using

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta_W \frac{\partial E}{\partial \boldsymbol{W}} \quad \text{and} \quad \boldsymbol{H} \leftarrow \boldsymbol{H} - \eta_H \frac{\partial E}{\partial \boldsymbol{H}} \tag{5}$$

where η_W and η_H are step sizes which, if chosen cleverly, guarantee that any intermediate solutions for $\boldsymbol{W}$ and $\boldsymbol{H}$ remain non-negative [11].

As of this writing, numerous variations of these two ideas have been proposed which, for instance, involve projected- or sub-gradient methods [12, 15]. Further details and theoretical properties regarding such approaches can be found in [5].

We conclude our discussion of the properties of NMF by noting that solutions found through iteratively solving (1) critically depend on how $\boldsymbol{W}$ and $\boldsymbol{H}$ are initialized [3]. In fact, solutions found from considering (1) are usually not unique [10]. This can easily bee seen as follows: Let $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$ and let $\boldsymbol{D}$ be a scaling matrix, then $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{D}\boldsymbol{D}^{-1}\boldsymbol{H} = \tilde{\boldsymbol{W}}\tilde{\boldsymbol{H}}$ which is to say that NMF “suffers” from indeterminate scales. Our discussion below will clarify this claim.



(a) non-negative data and convex cone

(b) data points defining edges

Fig. 2: Illustration of the fact that non-negative data reside in a polyhedral cone.

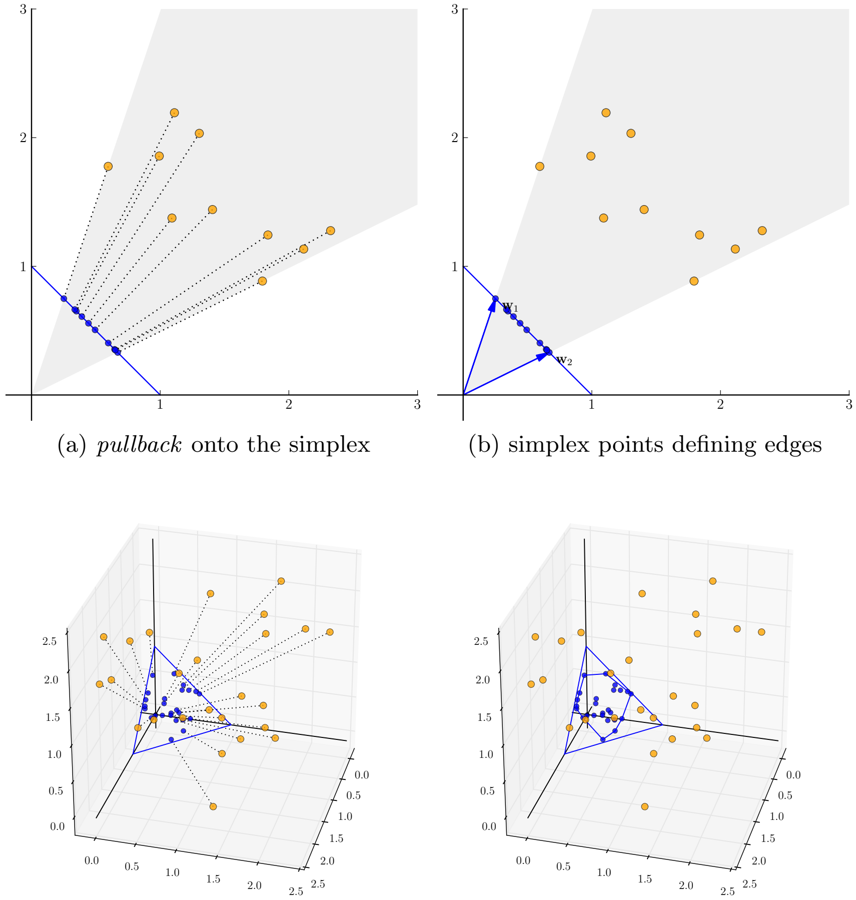
3 The Geometry of NFM

In the context of NMF, Donoho and Stodden [7] were the first to point to the fact that any set of non-negative vectors of arbitrary dimensionality resides within a convex cone which itself is embedded in the positive orthant of the corresponding vector space (see Fig. 2(a) for 2-dimensional example).

Since practical applications usually deal with finitely many data points, we note that any finite set of non-negative vectors $\boldsymbol{x}_j \in \mathbb{R}^m$ lies indeed within a *convex polyhedral cone*, i.e. within the convex hull of a set of halflines whose directions are defined by some of the given vectors. This is illustrated in Fig. 2(b) where the two vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ that define the edges of the cone coincide with two of the data points.

These observations hint at NMF approaches where the estimation of $\boldsymbol{W}$ can be *decoupled* from the computation of the coefficient matrix $\boldsymbol{H}$. If it was possible to identify those $p \leq n$ data points in $\boldsymbol{X}$ that define the edges of the enclosing polyhedral cone, they could either be used to perfectly reconstruct the data or we could select $k \leq p$ of them that would allow for reasonably good approximations. These prototypes would form $\boldsymbol{W}$ and the coefficient matrix $\boldsymbol{H}$ could be computed subsequently. Moreover, as shown in [18], such a decoupling would enable parallel NMF: Once $\boldsymbol{W}$ had been determined, the data matrix $\boldsymbol{X}$ could be partitioned into r blocks $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_r]$ where $\boldsymbol{X}_i \in \mathbb{R}^{m \times n/r}$. For each block, we could then solve (1) for the corresponding $\boldsymbol{H}_i$ which might be done on r cores simultaneously.

While the work in [18] approached the selection of suitable prototypes $\boldsymbol{w}_i$ from $\boldsymbol{X}$ by means of random projections, Chu and Lin [4] pointed out another interesting geometric property of NMF which we illustrate in Fig. 3. It shows that the cone that encloses the data in $\boldsymbol{X}$ remains invariant under certain simple



(a) *pullback* onto the simplex (b) simplex points defining edges

(c) data and pullback onto the simplex (d) convex hull on the simplex

Fig. 3: Pulling non-negative data back onto the standard simplex leaves the geometry of the enclosing polyhedral cone intact.

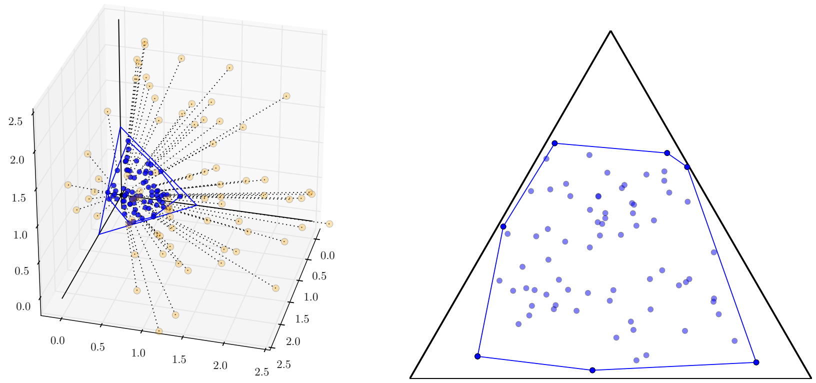
transformations. In particular, the so called *pullback*

$$\boldsymbol{y}_j = \frac{\boldsymbol{x}_j}{\sum_i x_{ij}} \tag{6}$$

which maps each data point $\boldsymbol{x}_j \in \mathbb{R}^m$ to a point $\boldsymbol{y}_j$ in the standard simplex Δ^{m-1} does not affect the halflines that define the cone.

Moreover, data points $\boldsymbol{x}_j$ on the edges of the cone in $\mathbb{R}^m$ will be mapped to vertices of the convex hull of the $\boldsymbol{y}_j \in \Delta^{m-1}$ (see Fig. 3). This observation is crucial, because it suggests that:

> The problem of estimating a suitable basis matrix $\boldsymbol{W}$ for NMF can be cast as a problem of *archetypal analysis* on the simplex.



(a) 3D data and simplex projection (b) convex hull on the simplex

Fig. 4: Pullback to the simplex and data convex hull on the simplex.

Archetypal analysis is a latent factor model due to Cutler and Breiman [6] who proposed to represent data in terms of convex combinations of extremes, that is, in terms of convex combinations of points on the convex hull of a given set of data. Recently, it spawned considerable research because it was recognized that it allows for a decoupled and thus efficient computation of basis elements and coefficients [2, 8, 13]. Next, we apply what we just established and combine our geometric considerations with approaches to efficient archetypal analysis so as to devise an NMF algorithm that notably differs from the techniques above.

4 Yet Another NMF Algorithm

Due to its practical utility, research on NMF has produced vast literature. Yet, except for only a few contributions (most notably [4, 10]), most NMF algorithms to date vary the ideas in [11, 14]. Our approach in this section, however, does not involve constrained optimization. It is related to the work in [4, 10] which apply geometric criteria to find suitable basis vectors. We extend these ideas in that we consider basis selection heuristics recently developed for archetypal analysis and demonstrate that NMF coefficients, too, can be computed without constrained optimization.

Above, we saw that optimal NMF is an NP hard problem. We further saw that traditional algorithms attempt to determine matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ simultaneously but that the geometry of non-negative data allows for a decoupled estimation of both matrices. While it is comparatively simple to determine coefficients once basis vectors are available, the difficulty lies in finding suitable basis vectors. We therefore first discuss estimating $\boldsymbol{W}$ and then address the task of computing $\boldsymbol{H}$.

4.1 Computing Matrix $\boldsymbol{W}$

In order to compute suitable basis vectors for a non-negative factorization of a given data set $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^n, \boldsymbol{x}_j \in \mathbb{R}^m$, we first transform the data using (6) and obtain a set of stochastic vectors $\mathcal{Y} = \{\boldsymbol{y}_j\}_{j=1}^n, \boldsymbol{y}_j \in \Delta^{m-1}$. Figure 4 illustrates this step by means of an examples of 3-dimensional data.

We note again that if we could determine the vertices $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_p$ of the convex hull of $\mathcal{Y}$ where $p \leq n$, we could perfectly reconstruct the given data as

$$\boldsymbol{x}_j = \sum_{i=1}^{p} h_{ij} \boldsymbol{w}_i, \quad h_{ij} \geq 0 \; \forall \; i. \tag{7}$$

However, in NMF we are interested in finding k basis vectors where k is usually chosen to be small. Yet, given the example in Fig. 4, we recognize that for higher dimensional data the convex hull of $\mathcal{Y}$ generally consists of many vertices so that p likely exceeds k. We are thus dealing with two problems: how to determine the vertices of $\mathcal{Y}$ and how to select k of them such that

$$\sum_{j=1}^{n} \left\| \boldsymbol{x}_j - \sum_{i=1}^{k} h_{ij} \boldsymbol{w}_i \right\|^2 \tag{8}$$

is as small as possible given that all the h_{ij} are non-negative?

These problems are indeed at the heart of recent work on archetypal analysis where it was shown that reasonable results can be obtained using the method of simplex volume maximization (SiVM) [17, 19] which answers both questions simultaneously. The idea is to select k points in $\mathcal{Y}$ that enclose a volume that is as large as possible. Given n points, it is easy to show that the $k \ll n$ points that enclose the largest volume will indeed be vertices of $\mathcal{Y}$.

Following the approach in [17], we apply *distance geometry* and note that the volume of a set of k vertices $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\} \subseteq \mathcal{Y}$ is given by

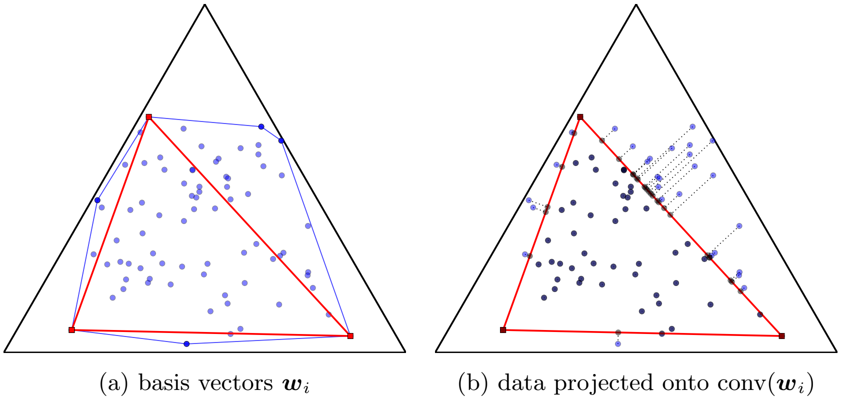
$$V^2(\mathcal{W}) = \frac{-1^k}{2^{k-1}\big((k-1)!\big)^2} \det(\boldsymbol{A}) \tag{9}$$

where

$$\det(\boldsymbol{A}) = \begin{vmatrix} 0 & 1 & 1 & 1 & \ldots & 1 \\ 1 & 0 & d_{11}^2 & d_{12}^2 & \ldots & d_{1k}^2 \\ 1 & d_{11}^2 & 0 & d_{22}^2 & \ldots & d_{2k}^2 \\ 1 & d_{12}^2 & d_{22}^2 & 0 & \ldots & d_{3k}^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{1k}^2 & d_{2k}^2 & d_{3k}^2 & \ldots & 0 \end{vmatrix}$$

is the Cayley-Menger Determinant whose elements indicate distance between the elements in $\mathcal{W}$ and are simply given by

$$d_{rs}^2 = \|\boldsymbol{w}_r - \boldsymbol{w}_s\|^2. \tag{10}$$



(a) basis vectors $\boldsymbol{w}_i$ (b) data projected onto conv($\boldsymbol{w}_i$)

Fig. 5: $k = 3$ basis vectors found by SiVM through greedy stochastic hill climbing and projections of data points onto the corresponding convex hull conv($\boldsymbol{w}_i$).

Algorithm 1 SiVM through greedy stochastic hill climbing

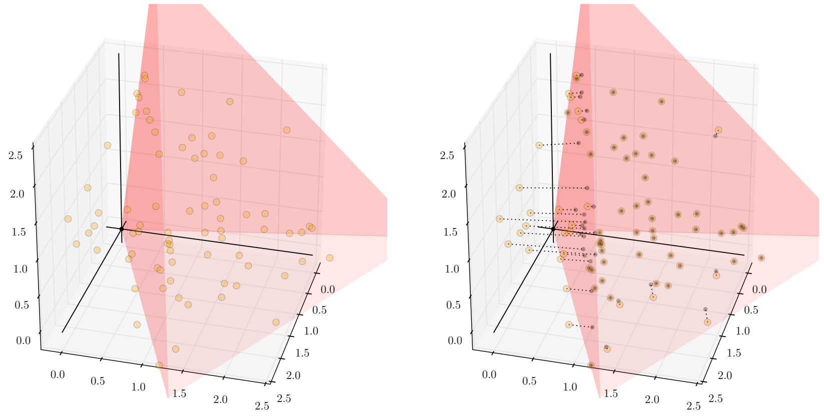
```
randomly select $\mathcal{W} \subset \mathcal{Y}$
for $\boldsymbol{y}_j \in \mathcal{Y}$ do
    for $\boldsymbol{w}_i \in \mathcal{W}$ do
        if $V\big(\mathcal{W} \setminus \{\boldsymbol{w}_i\} \cup \{\boldsymbol{y}_j\}\big) > V\big(\mathcal{W}\big)$ then
            $\mathcal{W} \leftarrow \mathcal{W} \setminus \{\boldsymbol{w}_i\} \cup \{\boldsymbol{y}_j\}$
```

We note again that NMF is an NP hard problem and that there is no free lunch. That is, even if we reduce the estimation of $\boldsymbol{W}$ to the problem of selecting suitable vertices in $\mathcal{Y}$, we are still dealing with a subset selection problem of the order of $\binom{n}{k}$. Aiming at efficiency, we resort to a greedy stochastic hill climbing variant of SiVM that was proposed in [9]. It initializes $\mathcal{W}$ by randomly selecting k points from $\mathcal{Y}$ then iterates over the $\boldsymbol{y}_j \in \mathcal{Y}$ and tests if replacing any of the $\boldsymbol{w}_i \in \mathcal{W}$ by $\boldsymbol{y}_j$ would lead to a larger volume. If so, the replacement is carried out and the search continues. Pseudocode of this procedure is shown in Algorithm 1 and Fig. 5(a) shows $k = 3$ basis vectors found in our example.

Concluding this subsection, we note that the basis vectors $\boldsymbol{w}_i$ determined from the simplex projected data $\boldsymbol{y}_j$ are all stochastic vectors whose entries are greater or equal than zero and sum to one. In contrast to conventional NMF approaches they are thus comparable in nature and do not suffer from ambiguous scales.

4.2 Computing Matrix $\boldsymbol{H}$

Once a set $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ of k basis vectors has been selected from the simplex projected data $\boldsymbol{y}_j$, every original data point $\boldsymbol{x}_j$ that lies within the



(a) polyhedral cone spanned by the $\boldsymbol{w}_i$ (b) data $\boldsymbol{x}_j$ and their projections $\hat{\boldsymbol{x}}_j$

Fig. 6: Non-negative data and polyhedral cone spanned by $k = 3$ basis vectors found through SiVM. If each $\boldsymbol{x}_j$ is projected to its closest point $\hat{\boldsymbol{x}}_j$ on this cone, it is easy to determine coefficients $h_i j \geq 0$ such that $\boldsymbol{x}_j \approx \hat{\boldsymbol{x}}_j = \sum_i h_{ij} \boldsymbol{w}_i$.

polyhedral cone spanned by the $\boldsymbol{w}_i$ can be perfectly reconstructed as

$$\boldsymbol{x}_j = \sum_{i=1}^{k} h_{ij} \boldsymbol{w}_i, \quad h_{ij} \geq 0 \, \forall \, i. \tag{11}$$

However, points outside that polyhedral cone cannot be expressed using non-negative coefficients. Typically, the best possible non-negative coefficients would therefore be determined using constrained least squares optimization. Here, we consider a different idea namely to project every $\boldsymbol{x}_j$ to its closest point in the polyhedral cone of the $\boldsymbol{w}_i$ and to determine coefficients for the projected point.

To achieve this, we first project the $\boldsymbol{y}_j$ onto the convex hull of the $\boldsymbol{w}_i$ in the simplex Δ^{m-1} and note that there are highly efficient computational geometry algorithms for this purpose [16, 22]. Figure 5(b) shows the corresponding result for our running example.

Let $\boldsymbol{z}_j$ denote the closest point of $\boldsymbol{y}_j$ in the convex hull of the $\boldsymbol{w}_i$. We then rescale the $\boldsymbol{z}_j$ to unit length, i.e.

$$\boldsymbol{z}_j \leftarrow \frac{\boldsymbol{z}_j}{\|\boldsymbol{z}_j\|} \tag{12}$$

and compute

$$\hat{\boldsymbol{x}}_j = \boldsymbol{z}_j \cdot \left(\boldsymbol{z}_j^T \boldsymbol{x}_j \right) \tag{13}$$

for all the original data vectors and thus obtain the point $\hat{\boldsymbol{x}}_j$ in the polyhedral cone of the $\boldsymbol{w}_i$ that is closest to $\boldsymbol{x}_j$. The corresponding result in our example can be seen in Fig. 6 which shows the original data and their projections onto the polyhedral cone spanned by the $k = 3$ basis vectors found previously.

The $\hat{\boldsymbol{x}}_j$ are then gathered in a matrix $\hat{\boldsymbol{X}}$ and a unique coefficient matrix $\boldsymbol{H}$ that, by nature of the $\hat{\boldsymbol{x}}_j$, will only contain non-negative entries is computed as

$$\boldsymbol{H} = \left(\boldsymbol{W}^T\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\hat{\boldsymbol{X}} \tag{14}$$

so that we indeed obtain two factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ for which $\boldsymbol{W}\boldsymbol{H} \approx \boldsymbol{X}$.

5 Conclusion

In this paper, we first discussed traditional approaches to non-negative matrix factorization and pointed out some of the difficulties that arise in this context. We then assumed a geometric point of view on the problem and showed in a step by step construction that it is possible to compute NMF of a data matrix without having to resort to sophisticated methods from optimization theory.

We believe that there are several advantages to our approach. First of all, it is computationally simple and allows for parallelization. Second of all, it is intuitive and easy to visualize and thus provides alternative avenues for teaching this material to students. Third of all, it also creates new perspectives for NMF research. While traditional, optimization-based approaches to NMF are very well understood by now and most related recent publications are but mere variations of a common theme, the idea of matrix factorization as search for suitable basis vectors by means of geometric objectives such as maximum volumes raises new questions. For instance, in ongoing work we are currently exploring the role of *entropy* in NMF. Given the pullback onto the simplex, it is obvious to consider the entropy of the resulting stochastic vectors as a criterion for their selection as possible basis vectors. Indeed, points with lower entropy are situated closer to the simplex boundary and therefore seem appropriate candidates for basis vectors. Corresponding search algorithms are under development and we hope to report results soon.

References

1. Aloise, D., Deshapande, A., Hansen, P., Popat, P.: NP-Hardness of Euclidean Sum-of-Squares Clustering. Machine Learning 75(2), 245–248 (2009)
2. Bauckhage, C., Thurau, C.: Making Archetypal Analysis Practical. In: Pattern Recogntion. LNCS, vol. 5748, pp. 272–281. Springer (2009)
3. Berry, M., Browne, M., Langville, A., Pauca, V., Plemmons, R.: Algorithms and Applications for Approximate Nonnegative Matrix Factorization. Computational Statistics and Data Analysis 52(1), 155–173 (2007)
4. Chu, M., Lin, M.: Low-Dimensional Polytope Approximation and Its Applications to Nonnegative Matrix Factorization. SIAM J. on Scientific Computing 30(3), 1131–1155 (2008)
5. Cichocki, A., Zdunek, R., Phan, A., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley (2009)
6. Cutler, A., Breiman, L.: Archetypal Analysis. Technometrics 36(4), 338–347 (1994)

7. Donoho, D., Stodden, V.: When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts? In: Proc. NIPS (2003)
8. Eugster, M., Leisch, F.: Weighted and Robust Archetypal Analysis. Computational Statistics & Data Analysis 55(3), 1215–1225 (2011)
9. Kersting, K., Bauckhage, C., Thurau, C., Wahabzada, M.: Matrix Factorization as Search. In: Proc. Eur. Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2012)
10. Klingenberg, B., Curry, J., Dougherty, A.: Non-negative Matrix Factorization: Ill-posedness and a Geometric Algorithm. Pattern Recognition 42(5), 918–928 (2008)
11. Lee, D., Seung, S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature 401(6755), 788–791 (1999)
12. Lin, C.J.: Projected Gradient Methods for Non-negative Matrix Factorization. Neural Computation 19(10), 2756–2779 (2007)
13. Morup, M., Hansen, L.: Archetypal Analysis for Machine Learning and Data Mining. Neurocomputing 80, 54–63 (2012)
14. Paatero, P., Tapper, U.: Positive Matrix Factorization: A Non-negative Factor Model with Optimal Utilization of Error Estimates of Data Values. Environmetrics 5(2), 11–126 (1994)
15. Recht, B., Re, C., Tropp, J., Bittorf, V.: Factoring Nonnegative Matrices with Linear Programs. In: Proc. NIPS (2012)
16. Sekitani, K., Yamamoto, Y.: A Recursive Algorithm for Finding the Minimum Norm Point in a Polytope and a Pair of Closest Points in Two Polytopes. Mathematical Programming 61(1), 233–249 (1993)
17. Thurau, C., Kersting, K., Bauckhage, C.: Yes We Can – Simplex Volume Maximization for Descriptive Web-Scale Matrix Factorization. In: Proc. Int. Conf. on Information and Knowledge Management. ACM (2010)
18. Thurau, C., Kersting, K., Wahabzada, M., Bauckhage, C.: Convex Non-negative Matrix Factorization for Massive Datasets. Knowledge and Information Systems 29(2), 457–478 (2011)
19. Thurau, C., Kersting, K., Wahabzada, M., Bauckhage, C.: Descriptive Matrix Factorization for Sustainability: Adopting the Principle of Opposites. Data Mining and Knowledge Discovery 24(2), 325–354 (2012)
20. Vasiloglou, N., Gray, A., Anderson, D.: Non-Negative Matrix Factorization, Convexity and Isometry. In: Proc. SIAM Int. Conf. on Data Mining (2009)
21. Vavasis, S.: On the Complexity of Nonnegative Matrix Factorization. SIAM J. on Optimization 20(3), 1364–1377 (2009)
22. Wolfe, P.: Finding the Nearest Point in a Polytope. Mathematical Programming 11(1), 128–149 (1976)