k-Means Clustering via the Frank-Wolfe Algorithm

Christian Bauckhage

B-IT, University of Bonn, Bonn, Germany
Fraunhofer IAIS, Sankt Augustin, Germany
http://multimedia-pattern-recognition.info

Abstract. We show that k-means clustering is a matrix factorization problem. Seen from this point of view, k-means clustering can be computed using alternating least squares techniques and we show how the constrained optimization steps involved in this procedure can be solved efficiently using the Frank-Wolfe algorithm.

1 Introduction

In this paper, we are concerned with theoretical aspects of machine learning. In particular, we revisit k-means clustering and investigate it from the point of view of data matrix factorization.

The k-means procedure is a popular technique to cluster a data set X of numerical data into subsets $C_1, \ldots, C_k$. The underlying ideas are intuitive and simple and most properties of k-means clustering are text book material [1, 2]. Adding to this material, several authors have recently argued that k-means clustering can be understood as a constrained matrix factorization problem [3–7]. However, reading the related literature, one cannot but notice that most authors consider this fact self explanatory and appeal to intuition.

Our goals with this paper are therefore as follows: i) we provide a rigorous proof for the equivalence of k-means clustering and constrained data matrix factorization. ii) we show that the matrix factorization point of view immediately reveals several properties of k-means clustering which are usually difficult to work out. In particular, we show that k-means clustering is an integer programming problem and therefore NP-hard, that k-means clustering allows for invoking the kernel trick, and that k-means clustering is closely related to archetypal analysis [8, 9] and non-negative matrix factorization [10, 11]. iii) we show that the matrix factorization perspective leads to yet another algorithm for computing k-means clustering and we discuss how to efficiently implement it using the Frank-Wolfe optimization scheme [12–14].

We begin our presentation with a brief summery of the traditional view on k-means clustering and then move on to the matrix factorization perspective. Our discussion assumes that readers are familiar with theory and practice of matrix factorization for data mining and machine learning. Those interested in a gentle introduction into the underlying mathematical ideas are referred to [11].

2 *k*-Means Clustering: Known Properties and Algorithms

Given a set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of data points $\boldsymbol{x}_j \in \mathbb{R}^m$, *hard k-means clustering* attempts to partition the data into k clusters $C_1, \ldots, C_k$ such that $C_i \subset X$, $C_i \cap C_l = \emptyset$, and $C_1 \cup C_2 \cup \ldots \cup C_k = X$. In particular, hard k-means clustering is a prototype-based clustering technique because it understands clusters to be defined in terms of prototypes or *cluster centroids* $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^m$, namely

$$C_i = \left\{ \boldsymbol{x}_j \in X \;\middle|\; \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2 \leq \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_l\right\|^2 \, \forall \, l \neq i \right\} \tag{1}$$

The problem at the heart of hard k-means clustering is therefore to search for k appropriate cluster centroids which are typically determined as the minimizers of the following objective function

$$E(k) = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in C_i} \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2 = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2 \tag{2}$$

where the

$$z_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{x}_j \in C_i \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

are binary indicator variables which indicate whether or not data point $\boldsymbol{x}_j$ belongs to cluster C_i.

Since hard k-means clustering aims at disjoint clusters where each $\boldsymbol{x}_j$ is assigned to one and only one C_i, we point out the following important properties of the $z_{ij} \in \{0, 1\}$. If we fix the data index j and sum over the cluster index i, we obtain the number of clusters data point $\boldsymbol{x}_j$ is assigned to, namely

$$\sum_{i=1}^{k} z_{ij} = 1. \tag{4}$$

Also, by fixing the cluster index i and summing over the data index j, we find

$$\sum_{j=1}^{n} z_{ij} = |C_i| = n_i \tag{5}$$

where n_i indicates the number of data points assigned to cluster C_i.

Although the objective in (2) looks rather innocent, it is actually NP-hard [15] and has to be approached using heuristics for which there is no guarantee to find the optimal solution. Indeed, there are various k-means heuristics or algorithms of which well known examples include Lloyd's algorithm (aka "the" k-means algorithm) [16], Hartigan's algorithm [17–19], MacQueen's algorithm [20], or gradient descend methods [21].

3 k-Means Clustering Is Data Matrix Factorization

Having recalled properties of hard k-means clustering in the previous section, our goal is now to rigorously establish that k-means clustering is matrix factorization. In other words, given the objective in (2), we will prove the following identity

$$\sum_{i=1}^{k}\sum_{j=1}^{n} z_{ij} \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2 = \left\|\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\right\|^2 \tag{6}$$

where

$$\boldsymbol{X} \in \mathbb{R}^{m\times n} \text{ is a column matrix of } n \text{ data vectors } \boldsymbol{x}_j \in \mathbb{R}^m \tag{7}$$

$$\boldsymbol{M} \in \mathbb{R}^{m\times k} \text{ is a column matrix of } k \text{ cluster centroids } \boldsymbol{\mu}_i \in \mathbb{R}^m \tag{8}$$

$$\boldsymbol{Z} \in \mathbb{R}^{k\times n} \text{ is a matrix of binary indicator variables such that}$$

$$z_{ij} = \begin{cases} 1, \text{ if } \boldsymbol{x}_j \in C_i \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

3.1 Notation and Preliminaries

Throughout, we write $\boldsymbol{x}_j$ to denote j-th column vector of a matrix $\boldsymbol{X}$. To refer to the (l, j) element of a matrix $\boldsymbol{X}$, we either write x_{lj} or $\big(\boldsymbol{X}\big)_{lj}$. Moreover, subscripts or summation indices i will be understood to range from 1 to k (the number of clusters), subscripts or summation indices j will range from 1 up to n (the number of data), and subscripts or summation indices l will be used to expand inner products between vectors or rows and columns of matrices.

Finally, regarding the squared Frobenius norm of a matrix, we recall that

$$\left\|\boldsymbol{X}\right\|^2 = \sum_{l,j} x_{lj}^2 = \sum_{j} \left\|\boldsymbol{x}_j\right\|^2 = \sum_{j} \boldsymbol{x}_j^T\boldsymbol{x}_j = \sum_{j} \big(\boldsymbol{X}^T\boldsymbol{X}\big)_{jj} = \operatorname{tr}\big[\boldsymbol{X}^T\boldsymbol{X}\big] \tag{10}$$

3.2 Step by Step Derivation of (6)

To substantiate the claim in (6), we first point out a crucial property of the binary indicator matrix $\boldsymbol{Z}$ in (9). The property of the $z_{ij} \in \{0, 1\}$ we worked out in (4) translates to the statement that each column of $\boldsymbol{Z}$ contains a single entry of 1 and $k - 1$ entries of 0. This immediately establishes that the rows of $\boldsymbol{Z}$ are pairwise perpendicular because

$$z_{ij}\, z_{i'j} = \begin{cases} 1, & \text{if } i = i' \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

which is then to say that the matrix $\boldsymbol{Z}\boldsymbol{Z}^T$ is a diagonal matrix where

$$\big(\boldsymbol{Z}\boldsymbol{Z}^T\big)_{ii'} = \sum_{j} \big(\boldsymbol{Z}\big)_{ij}\big(\boldsymbol{Z}^T\big)_{ji'} = \sum_{j} z_{ij}\, z_{i'j} = \begin{cases} n_i, & \text{if } i = i' \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Having discussed this property of $\boldsymbol{Z}$, we are now positioned to establish the equality in (6) and we will do this in a step by step manner.

Step 1: Expanding the expression on the left of (6) First, we expand the conventional k-means objective function and find

$$\begin{aligned}\sum_{i,j} z_{ij} \left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2 &= \sum_{i,j} z_{ij} \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{x}_j^T \boldsymbol{\mu}_i + \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i\right) \\ &= \underbrace{\sum_{i,j} z_{ij}\, \boldsymbol{x}_j^T \boldsymbol{x}_j}_{T_1} - 2 \underbrace{\sum_{i,j} z_{ij}\, \boldsymbol{x}_j^T \boldsymbol{\mu}_i}_{T_2} + \underbrace{\sum_{i,j} z_{ij}\, \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i}_{T_3}. \end{aligned} \tag{13}$$

This expansion leads to further insights, if we examine the three terms T_1, T_2, and T_3 one by one. First of all, we find

$$T_1 = \sum_{i,j} z_{ij}\, \boldsymbol{x}_j^T \boldsymbol{x}_j = \sum_{i,j} z_{ij} \left\|\boldsymbol{x}_j\right\|^2 = \sum_j \left\|\boldsymbol{x}_j\right\|^2 \sum_i z_{ij} = \sum_j \left\|\boldsymbol{x}_j\right\|^2 \tag{14}$$

$$= \operatorname{tr}\big[\boldsymbol{X}^T \boldsymbol{X}\big] \tag{15}$$

where we made use of (4) and (10). Second of all, we observe

$$T_2 = \sum_{i,j} z_{ij}\, \boldsymbol{x}_j^T \boldsymbol{\mu}_i = \sum_{i,j} z_{ij} \sum_l x_{lj}\, \mu_{li} \tag{16}$$

$$= \sum_{j,l} x_{lj} \sum_i \mu_{li}\, z_{ij} \tag{17}$$

$$= \sum_{j,l} x_{lj} \left(\boldsymbol{M}\boldsymbol{Z}\right)_{lj} \tag{18}$$

$$= \sum_{j,l} \left(\boldsymbol{X}^T\right)_{jl} \left(\boldsymbol{M}\boldsymbol{Z}\right)_{lj} \tag{19}$$

$$= \sum_j \left(\boldsymbol{X}^T \boldsymbol{M}\boldsymbol{Z}\right)_{jj} \tag{20}$$

$$= \operatorname{tr}\big[\boldsymbol{X}^T \boldsymbol{M}\boldsymbol{Z}\big] \tag{21}$$

Third of all, we note that

$$T_3 = \sum_{i,j} z_{ij}\, \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i = \sum_{i,j} z_{ij} \left\|\boldsymbol{\mu}_i\right\|^2 = \sum_i \left\|\boldsymbol{\mu}_i\right\|^2 \sum_j z_{ij} = \sum_i \left\|\boldsymbol{\mu}_i\right\|^2 n_i \tag{22}$$

where we applied (5).

Step 2: Expanding the expression on the right of (6) Next, we look at the squared Frobenius norm on the right hand side of (6); it can be written as

$$\begin{aligned}\left\|\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\right\|^2 &= \operatorname{tr}\Big[\left(\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\right)^T \left(\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\right)\Big] \\ &= \underbrace{\operatorname{tr}\big[\boldsymbol{X}^T \boldsymbol{X}\big]}_{T_4} - 2 \underbrace{\operatorname{tr}\big[\boldsymbol{X}^T \boldsymbol{M}\boldsymbol{Z}\big]}_{T_5} + \underbrace{\operatorname{tr}\big[\boldsymbol{Z}^T \boldsymbol{M}^T \boldsymbol{M}\boldsymbol{Z}\big]}_{T_6} \end{aligned} \tag{23}$$

Given our earlier results, we immediately recognize that $T_1 = T_4$ and $T_2 = T_5$. Thus, to establish that (13) and (23) are indeed equivalent, it remains to verify whether $T_3 = T_6$?

Regarding T_6, we note that, because of the cyclic permutation invariance of the trace operator, we have

$$\operatorname{tr}\big[\boldsymbol{Z}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{Z}\big] = \operatorname{tr}\big[\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{Z}^T\big]. \tag{24}$$

We also note that

$$\begin{aligned}
\operatorname{tr}\big[\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{Z}^T\big] &= \sum_i \big(\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{Z}^T\big)_{ii} && (25)\\
&= \sum_i \sum_l \big(\boldsymbol{M}^T\boldsymbol{M}\big)_{il}\big(\boldsymbol{Z}\boldsymbol{Z}^T\big)_{li} && (26)\\
&= \sum_i \big(\boldsymbol{M}^T\boldsymbol{M}\big)_{ii}\big(\boldsymbol{Z}\boldsymbol{Z}^T\big)_{ii} && (27)\\
&= \sum_i \big\|\boldsymbol{\mu}_i\big\|^2 n_i && (28)
\end{aligned}$$

where we used the fact that $\boldsymbol{Z}\boldsymbol{Z}^T$ is diagonal. This result, however, shows that $T_3 = T_6$ and, consequently, that (13) and (23) are equivalent. That is, we have proven that k-means clustering can indeed be cast as a matrix factorization problem.

4 Consequences

Having proved our central claim in (6), we will next discuss several consequences we can obtain from the matrix factorization formulation of k-means clustering.

4.1 k-Means Clustering Is NP Hard

Given the above result, further insights into the nature of k-means clustering result from eliminating matrix $\boldsymbol{M}$ from the right hand side of (6). That is, we next ask for the matrix $\boldsymbol{M}$ that, for a given $\boldsymbol{Z}$, would minimize $\big\|\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\big\|^2$. To this end, we consider

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{M}}\Big\|\boldsymbol{X} - \boldsymbol{M}\boldsymbol{Z}\Big\|^2 &= \frac{\partial}{\partial \boldsymbol{M}}\Big[\operatorname{tr}\big[\boldsymbol{X}^T\boldsymbol{X}\big] - 2\operatorname{tr}\big[\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{Z}\big] + \operatorname{tr}\big[\boldsymbol{Z}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{Z}\big]\Big]\\
&= 2\big(\boldsymbol{M}\boldsymbol{Z}\boldsymbol{Z}^T - \boldsymbol{X}\boldsymbol{Z}^T\big) && (29)
\end{aligned}$$

which, upon equating to $\boldsymbol{0}$, leads to

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Z}^T\big(\boldsymbol{Z}\boldsymbol{Z}^T\big)^{-1} \tag{30}$$

which beautifully reflects the fact that each of the k-means cluster centroids $\boldsymbol{\mu}_i$ coincides with the mean of the corresponding cluster C_i, namely

$$\boldsymbol{\mu}_i = \frac{\sum_j z_{ij}\, \boldsymbol{x}_j}{\sum_j z_{ij}} = \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j. \tag{31}$$

Given the result in (30), we therefore find that the k-means objective can be cast solely in terms of the data matrix $\boldsymbol{X}$ and the indicator matrix $\boldsymbol{Z}$

$$\begin{aligned} \min_{\boldsymbol{Z}} \quad & \Big\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^T \big(\boldsymbol{Z}\boldsymbol{Z}^T\big)^{-1} \boldsymbol{Z} \Big\|^2 \\ \text{s.t.} \quad & z_{ij} \in \{0,1\} \\ & \sum_i z_{ij} = 1 \end{aligned} \tag{32}$$

Looking at (32), we recognize k-means clustering as an integer programming problem, since it corresponds to the discrete optimization problem of finding a column stochastic binary matrix $\boldsymbol{Z}$ that minimizes the objective in (32). Integer programming problems are NP hard and we can actually read this off (32). $\boldsymbol{Z}$ is an $k \times n$ binary matrix such that each column contains a single 1; thus, for each column there are k ways to place that 1 and since there are n columns, there are $O\big(k^n\big)$ matrices among which we have to determine the optimal one.

4.2 Kernel k-Means Clustering

Observe that the squared Frobenius norm in (32) can also be expanded in terms of trace operators. If we substitute $\boldsymbol{\Xi} = \boldsymbol{Z}^T \big(\boldsymbol{Z}\boldsymbol{Z}^T\big)^{-1} \boldsymbol{Z}$ for brevity, we find

$$\Big\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{\Xi} \Big\|^2 = \operatorname{tr}\Big[\boldsymbol{X}^T\boldsymbol{X}\Big] - 2\operatorname{tr}\Big[\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\Xi}\Big] + \operatorname{tr}\Big[\boldsymbol{\Xi}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\Xi}\Big] \tag{33}$$

and recognize that each occurrence of the data vectors $\boldsymbol{x}_i$ is in form of an inner product, because $\big(\boldsymbol{X}^T\boldsymbol{X}\big)_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j$.

This, however, shows that k-means allows for invoking the kernel trick as we may replace the $n \times n$ Gramian $\boldsymbol{X}^T\boldsymbol{X}$ by an $n \times n$ kernel matrix $\boldsymbol{K}$ whose entries correspond to kernel evaluations $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

4.3 k-Means Clustering, Archetypal Analysis and Non-Negative Matrix Factorization

Even further insights into the nature of k-means clustering arise, if we substitute $\boldsymbol{Y} = \boldsymbol{Z}^T \big(\boldsymbol{Z}\boldsymbol{Z}^T\big)^{-1}$ so that we can write $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}$. This again reveals that the centroid vectors $\boldsymbol{\mu}_i$, i.e. the columns of $\boldsymbol{M}$, are convex combinations of data points $\boldsymbol{x}_j$. That is, $\boldsymbol{\mu}_i = \boldsymbol{X}\boldsymbol{y}_i$ where $\boldsymbol{y}_i$ is an n dimensional vector with n_i entries equal to $1/n_i$ and $n - n_i$ entries equal to 0.

We note that, by definition, $\boldsymbol{Y}$ is a column stochastic matrix. It is non-negative, i.e. $\boldsymbol{Y} \succeq \boldsymbol{0}$, and its columns sum to one, i.e. $\sum_j y_{ji} = \boldsymbol{1}^T \boldsymbol{y}_i = 1$. Moreover, each column will have high entropy $H(\boldsymbol{y}_i) = -\sum_j y_{ji} \log y_{ji} \gg 0$.

We also recall that, as a binary matrix, matrix $\boldsymbol{Z}$ will obey $\boldsymbol{Z} \succeq \boldsymbol{0}$, $\boldsymbol{1}^T \boldsymbol{z}_j = 1$, and $H(\boldsymbol{z}_j) = -\sum_i z_{ij} \log z_{ij} = 0$.

Hard k-Means Clustering Given these prerequisites, we can therefore express hard k-means clustering as a constrained quadratic optimization problem

$$\begin{aligned} \min_{\boldsymbol{Y},\boldsymbol{Z}} \quad & \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z} \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{Y} \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{y}_i = 1,\ H(\boldsymbol{y}_i) \gg 0 \\ & \boldsymbol{Z} \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{z}_j = 1,\ H(\boldsymbol{z}_j) = 0. \end{aligned} \tag{34}$$

Archetypal Analysis If we drop the entropy constraints in (34), we obtain

$$\begin{aligned} \min_{\boldsymbol{Y},\boldsymbol{Z}} \quad & \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z} \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{Y} \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{y}_i = 1 \\ & \boldsymbol{Z} \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{z}_j = 1 \end{aligned} \tag{35}$$

and recover a problem known as *archetypal analysis* (AA) [8, 9]. Dropping the entropy constraints has an interesting effect. Instead of computing basis vectors $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}$ that correspond to local means, AA determines basis vectors that are extreme points of the data in $\boldsymbol{X}$. In fact, the *archetypes* in matrix $\boldsymbol{M}$ reside on the data convex hull [22].

Non-Negative Matrix Factorization If we further drop the sum-to-one constraints in (35), we obtain

$$\begin{aligned} \min_{\boldsymbol{Y},\boldsymbol{Z}} \quad & \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z} \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{Y} \succeq \boldsymbol{0} \\ & \boldsymbol{Z} \succeq \boldsymbol{0} \end{aligned} \tag{36}$$

a problem known as *non-negative matrix factorization* (NMF) [10]. Dropping the stochastic constraints has the effect that NMF computes basis vectors $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}$ that are conic combinations of the data in $\boldsymbol{X}$.

The expressions in (34), (35), and (36) therefore reveal k-means clustering to be a particularly severely constrained quadratic optimization problem. This is interesting in so far as algorithms for computing k-means clustering conceptually much simpler than AA or NMF algorithms. Nevertheless, it now appears as if methods that have been developed for AA and NMF might also apply to k-means clustering. In the next section, we show that this is indeed the case.

5 Yet Another Algorithm for k-Means Clustering

Since we just found that hard k-means clustering is indeed a constrained form of archetypal analysis, the question is if algorithms that have been developed for archetypal analysis can be used to compute k-means clustering.

In order to see how this can be accomplished, we note that the objective function in (34) is convex in either $\boldsymbol{Y}$ or $\boldsymbol{Z}$ but not in the their product $\boldsymbol{YZ}$. An idea for solving the problem in (34) could therefore be to apply the following constrained alternating least squares procedure

1) randomly initialize $\boldsymbol{Y}$ and $\boldsymbol{Z}$ under the appropriate constraints
2) fix matrix $\boldsymbol{Z}$ and update $\boldsymbol{Y}$ to the solution of

$$\begin{aligned} &\min_{\boldsymbol{Y}} \left\| \boldsymbol{X} - \boldsymbol{XYZ} \right\|^2 \\ &\text{s.t. } \boldsymbol{y}_i \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{y}_i = 1,\ H(\boldsymbol{y}_i) \gg 0 \end{aligned}$$

3) fix matrix $\boldsymbol{Y}$ and update $\boldsymbol{Z}$ to the solution of

$$\begin{aligned} &\min_{\boldsymbol{Z}} \left\| \boldsymbol{X} - \boldsymbol{XYZ} \right\|^2 \\ &\text{s.t. } \boldsymbol{z}_i \succeq \boldsymbol{0},\ \boldsymbol{1}^T \boldsymbol{z}_j = 1,\ H(\boldsymbol{z}_j) = 0 \end{aligned}$$

4) if not converged, continue at 2)

for which we note that the seemingly difficult constrained quadratic optimization problems it involves can actually be solved rather easily.

First of all, we observe that the stochastic constraints $\boldsymbol{y}_i \succeq \boldsymbol{0}$ and $\boldsymbol{1}^T \boldsymbol{y}_i = 1$ require the columns of matrix $\boldsymbol{Y}$ to reside in the standard n-simplex

$$\Delta^{n-1} = \left\{ \boldsymbol{y} \in \mathbb{R}^n \;\middle|\; \boldsymbol{y} \succeq \boldsymbol{0} \wedge \boldsymbol{1}^T \boldsymbol{y} = 1 \right\}. \tag{37}$$

Second of all, the column entropy $H(\boldsymbol{y}_i) = -\sum_j y_{ji} \log y_{ji}$ is a concave function so that $-H(\boldsymbol{y}_i)$ is convex and we are interested in solutions for $\boldsymbol{Y}$ that maximize $H(\boldsymbol{y}_i)$. We can thus rewrite the first problem in the above procedure as

$$\begin{aligned} \min_{\boldsymbol{Y}} \quad & \left\| \boldsymbol{X} - \boldsymbol{XYZ} \right\|^2 - \sum_i H(\boldsymbol{y}_i) \\ \text{s.t.} \quad & \boldsymbol{y}_i \in \Delta^{n-1}. \end{aligned} \tag{38}$$

Written in this form, we recognize the problem as a convex minimization problem over a compact convex set. This, however, is to say that it can be tackled using the efficient Frank-Wolfe procedure [12]. The Frank-Wolfe algorithm shown in Alg. 1 solves problems of the form

$$\begin{aligned} &\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \\ &\text{s.t. } \boldsymbol{x} \in S \end{aligned} \tag{40}$$

Algorithm 1 Frank-Wolfe algorithm to solve problem such as in (40)

guess a feasible point $\boldsymbol{x}_0$
for $t = 0, \ldots, t_{\max}$ **do**
determine $\boldsymbol{s}_t$ by solving

$$\min_{\boldsymbol{s} \in S} \; \boldsymbol{s}^T \nabla f(\boldsymbol{x}_t) \tag{39}$$

update the learning rate $\gamma_t = \frac{2}{t+2}$
update the current estimate $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t \left(\boldsymbol{s}_t - \boldsymbol{x}_t \right)$

where $S \subset \mathbb{R}^m$ is a compact, convex set and $f : S \to \mathbb{R}$ is a convex function. The key idea is to compute $\boldsymbol{s} \in S$ that minimizes $\boldsymbol{s}^T \nabla f(\boldsymbol{x}_t)$ and to use sub-gradient updates $\boldsymbol{s} - \boldsymbol{x}_t$ which guarantee that the updates will never leave the feasible set. The efficiency of the algorithm stems from the fact that it turns a quadratic optimization problem into a series of linear optimization problems and we point out that the minimum of a linear function $\boldsymbol{s}^T \nabla f(\boldsymbol{x})$ over a compact convex set will be attained at a vertex of that set.

With respect to our problem of a matrix minimization problem, we note that the gradient ∇f we are concerned with is given by

$$\nabla_{\boldsymbol{Y}} \left(\left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z} \right\|^2 - \sum_i H(\boldsymbol{y}_i) \right) = 2 \left[\boldsymbol{X}^T \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z}\boldsymbol{Z}^T - \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{Z}^T \right] - \boldsymbol{L} \tag{41}$$

where the components of matrix $\boldsymbol{L}$ amount to

$$L_{ji} = \frac{\partial}{\partial y_{ji}} \sum_j y_{ji} \log y_{ji} = \log y_{ji} + 1 \tag{42}$$

We can use the columns of this gradient matrix $\boldsymbol{G}$ to update the columns $\boldsymbol{y}_i$ of $\boldsymbol{Y}$ according to the Frank-Wolfe algorithm. We also point out, the the compact convex set we are dealing with is the standard simplex Δ^{n-1} whose vertices are given by the standard basis vectors $\boldsymbol{e}_l \in \mathbb{R}^n$. The problem of finding the minimizer $\boldsymbol{s} \in \Delta^{n-1}$ thus simplifies to finding the basis vector $\boldsymbol{e}_l$ that minimizes $\boldsymbol{e}_l^T \boldsymbol{g}_i$ which is simply to determine the minimal entry g_{li} in each column $\boldsymbol{g}_i$ of $\boldsymbol{G}$. All in all, these consideration then lead to Alg. 2 for computing updates of matrix $\boldsymbol{Y}$.

Similar considerations apply to the problem of computing the updates of the indicator matrix $\boldsymbol{Z}$ and correspondingly lead to Alg. 3.

To conclude this discussion, we point out that the Frank-Wolfe procedure quickly achieves ϵ-approximations of the optimal solution that are provably sparse [13]. In fact, one can show that after t iterations the current estimate is $O(1/t)$ from the optimal solution [13] which provides a convenient criterion for choosing the number $t_{\max}$ of iterations to be performed. For further details on the Frank-Wolfe algorithms as well as for a recent excellent survey of projection-free convex optimization over compact convex sets, we refer to [14].

Algorithm 2 Frank-Wolfe procedure to compute $\boldsymbol{Y}$ whose columns are in Δ^{n-1}

Require: data matrix $\boldsymbol{X}$, indicator matrix $\boldsymbol{Z}$, and parameter $t_{\max} \in \mathbb{N}$
  $\boldsymbol{Y} \leftarrow [\boldsymbol{e}_1, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_1]$ where $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^n$ // initialize $n \times k$ matrix $\boldsymbol{Y}$
  $t \leftarrow 0$
  repeat
    $\boldsymbol{G} = 2\left[\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z}\boldsymbol{Z}^T - \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{Z}^T\right] - \boldsymbol{L}$ // compute gradient matrix
    for $i = 1, \ldots, k$ **do** // update columns $\boldsymbol{y}_i$ of $\boldsymbol{Y}$
      $i' = \operatorname{argmin}_l G_{li}$
      $\boldsymbol{z}_i \leftarrow \boldsymbol{z}_i + 2/(t+2) \cdot (\boldsymbol{e}_{i'} - \boldsymbol{z}_i)$
    $t \leftarrow t + 1$
  until updates "become small" or $t = t_{\max}$

Algorithm 3 Frank-Wolfe procedure to compute $\boldsymbol{Z}$ whose columns are in Δ^{k-1}

Require: data matrix $\boldsymbol{X}$, coefficient matrix $\boldsymbol{Y}$, and parameter $t_{\max} \in \mathbb{N}$
  $\boldsymbol{Z} \leftarrow [\boldsymbol{e}_1, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_1]$ where $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^k$ // initialize $k \times n$ matrix $\boldsymbol{Z}$
  $t \leftarrow 0$
  repeat
    $\boldsymbol{G} = 2\left[\boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z} - \boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{X}\right]$ // compute gradient matrix
    for $j = 1, \ldots, n$ **do** // update columns $\boldsymbol{z}_j$ of $\boldsymbol{Z}$
      $j' = \operatorname{argmin}_l G_{lj}$
      $\boldsymbol{z}_j \leftarrow \boldsymbol{z}_j + 2/(t+2) \cdot (\boldsymbol{e}_{j'} - \boldsymbol{z}_j)$
    $t \leftarrow t + 1$
  until updates "become small" or $t = t_{\max}$

6 Conclusion

In this paper, we were concerned with machine learning theory. In particular, we were concerned with theoretical aspects of k-means clustering.

First of all, we rigorously established that k-means clustering is a constrained data matrix factorization problem. Second of all, this insight allowed us to easily uncover several properties of k-means clustering that are otherwise more difficult to show. Third of all, the matrix factorization point of view on k-means clustering allowed us to reveal its connections to archetypal analysis and non-negative matrix factorization. Finally, given that k-means clustering can be understood as a constrained version of archetypal analysis, we discussed yet another algorithm for k-means clustering. Archetypal analysis is often computed using alternating least squares optimization and we showed how to adapt this idea to k-means clustering. In particular, we discussed that the seemingly difficult constrained optimization problems involved in this procedure can be solved using the efficient Frank-Wolfe procedure for convex optimization over compact convex sets.

Again, the work reported here is of mainly theoretical interest. What is particularly striking is that it established k-means clustering as a more constrained and thus more difficult problem than archetypal analysis or non-negative matrix factorization. Yet, at the same time, traditional algorithms for k-means clustering are considerably simpler than those for the latter problems. This can be

seen as a call to arms for it suggests that there may be simpler algorithms for these kind of problems as well. Indeed, techniques such as k-maxoids clustering [23] which were derived from k-means clustering indicate that, say, archetypal analysis should be solvable by simple algorithms, too.

References

1. MacKay, D.: Information Theory, Inference, & Learning Algorithms. Cambridge University Press (2003)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
3. Ding, C., He, X., Simon, H.: On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In: Proc. SDM, SIAM (2005)
4. Gaussier, E., Goutte, C.: Relations between PLSA and NMF and Implications. In: Proc. SIGIR, ACM (2005)
5. Kim, J., Park, H.: Sparse Nonnegative Matrix Factorization for Clustering. Technical Report GT-CSE-08-01, Georgia Institute of Technology (2008)
6. Arora, R., Gupta, M., Kapila, A., Fazel, M.: Similarity-based Clustering by Left-Stochastic Matrix Factorization. J. of Machine Learning Research **14**(Jul.) (2013)
7. Bauckhage, C., Drachen, A., Sifa, R.: Clustering Game Behavior Data. IEEE Trans. on Computational Intelligence and AI in Games **7**(3) (2015)
8. Cutler, A., Breiman, L.: Archetypal Analysis. Technometrics **36**(4) (1994)
9. Bauckhage, C., Thurau, C.: Making Archetypal Analysis Practical. In: Pattern Recogntion. Volume 5748 of LNCS., Springer (2009)
10. 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)
11. Bauckhage, C.: A Purely Geometric Approach to Non-Negative Matrix Factorization. In: Proc. KDML-LWA. (2014)
12. Frank, M., Wolfe, P.: An Algorithm for Quadratic Programming. Naval Research Logistics Quarterly **3**(1–2) (1956)
13. Clarkson, K.: Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. ACM Trans. on Algorithms **6**(4) (2010)
14. Jaggi, M.: Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. J. of Machine Learning Research **28**(1) (2013)
15. Aloise, D., Deshapande, A., Hansen, P., Popat, P.: NP-Hardness of Euclidean Sum-of-Squares Clustering. Machine Learning **75**(2) (2009)
16. Lloyd, S.: Least Squares Quantization in PCM. IEEE Trans. on Information Theory **28**(2) (1982)
17. Hartigan, J., Wong, M.: Algorithm AS 136: A k-Means Clustering Algorithm. J. of the Royal Statistical Society C **28**(1) (1979)
18. Slonim, N., Aharoni, E., Crammer, K.: Hartigan's k-Means Versus Lloyd's k-Means – Is It Time for a Change? In: Proc. IJCAI. (2013)
19. Telgarsky, M., Vattani, A.: Hartigan's Method: k-means Clustering without Voronoi. In: Proc. AISTATS. (2010)
20. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. Berkeley Symp. on Mathematical Statistics and Probability. (1967)

21. Bottou, L., Bengio, Y.: Convergence Properties of the K-Means Algorithms. In: Proc. NIPS. (1995)
22. Bauckhage, C.: A Note on Archetypal Analysis and the Approximation of Convex Hulls. arXiv:1410.0642 [cs.NA] (2014)
23. Bauckhage, C., Sifa, R.: k-Maxoids Clustering. In: Proc. KDML-LWA. (2015)