Available online at www.sciencedirect.com
ScienceDirect
Linear Algebra and its Applications 429 (2008) 1409–1424
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Structured nonnegative matrix factorization with applications to hidden Markov realization and clustering

Bart Vanluyten*, Jan C. Willems, Bart De Moor

K.U.Leuven, ESAT/SCD(SISTA), Kasteelpark Arenberg 10, B-3001 Heverlee-Leuven, Belgium

Received 11 March 2007; accepted 4 March 2008
Available online 9 June 2008
Submitted by T. Damm

Abstract

In this paper, we study the *structured nonnegative matrix factorization* problem: given a square, nonnegative matrix P, decompose it as $P = VAV^\top$ with V and A nonnegative matrices and with the dimension of A as small as possible. We propose an iterative approach that minimizes the Kullback–Leibler divergence between P and $VAV^\top$ subject to the nonnegativity constraints on A and V with the dimension of A given. The approximate structured decomposition $P \simeq VAV^\top$ is closely related to the approximate symmetric decomposition $P \simeq VV^\top$. It is shown that the approach for finding an approximate structured decomposition can be adapted to solve the symmetric decomposition problem approximately. Finally, we apply the nonnegative decomposition $VAV^\top$ to the hidden Markov realization problem and to the clustering of data vectors based on their distance matrix.

Keywords: Nonnegative matrix factorization; Kullback–Leibler divergence; Multiplicative update formulas; cp-Rank

1. Introduction

The nonnegative matrix factorization problem is a long-standing problem in linear algebra. It consists of decomposing a given (elementwise) nonnegative matrix M into a product VH of minimal inner dimension with V and H nonnegative. It can be shown that there exists a finite algorithm to compute the minimal inner dimension of an exact decomposition. However, the complexity bounds on the number of arithmetic/boolean operations that this algorithm requires,

* Corresponding author. Tel.: +32 16328666; fax: +32 16321986.
E-mail address: bart.vanluyten@esat.kuleuven.be (B. Vanluyten).

0024-3795/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2008.03.010

are non-polynomial. In [8] the approximate nonnegative matrix factorization problem has been introduced. The idea is to choose an inner dimension and compute a factorization VH which approximates M optimally. In [7,8] iterative update formulas for V and H are given that retain the nonnegativity of V and H and that decrease the Kullback–Leibler divergence between P and VH. In addition it is shown that, if the algorithm converges, it converges to a stationary point of the Kullback–Leibler divergence between P and VH. Convergence to a minimum is not guaranteed, as the algorithm may also converge to a saddle point. During recent years, further theoretic research [3,5] on the nonnegative factorization has been performed and the factorization started to be used in various engineering applications (image processing, text mining, etc.) [2,9].

In this paper, we introduce the approximate *structured nonnegative matrix factorization*. It consists of approximating a square, nonnegative matrix P into a product $VAV^\top$ with V and A nonnegative. Following the approach of [6], we prove that an optimal approximation $VAV^\top$ of P in the Kullback–Leibler divergence has the same element sum as P. This theorem allows us to search for an optimal decomposition in the space of column stochastic matrices V and matrices A with the same sum as the matrix P. As a consequence of this fact, we are able to prove update formulas for the decomposition $P \simeq VAV^\top$. When P is symmetric, we propose an adapted algorithm with A symmetric. We also propose iterative formulas to decompose a nonnegative, symmetric matrix P into a product $VV^\top$. It will also be shown that the structured decomposition can be used in several engineering applications, such as the hidden Markov realization problem and clustering.

In Section 2, we review the classical nonnegative matrix factorization problem and its approximate solution by means of iterative update formulas. In Section 3, we state the structured nonnegative matrix factorization problem and propose iterative update formulas to solve this problem approximately. Section 4 deals with the symmetric matrix factorization. In Section 5, we use the factorization to find an approximate hidden Markov model corresponding to given probabilities of output strings of length 2 (i.e. the hidden Markov realization problem for strings of length 2). In Section 6, we show that the factorization can be used for clustering data points based on the distance matrix between the points.

The following notation is used. $\mathbb{R}_+$ is the set of nonnegative real numbers. If X is a matrix, then we mean with X_{ij} the i, jth element of X, with $X_{i:}$, the ith row of X and with $X_{:j}$, the jth column of X. $X \geqslant 0$ denotes that the elements of X are nonnegative. With e we indicate a column vector with all elements equal to 1, i.e. $e = [1 \quad 1 \quad \cdots \quad 1]^\top$.

2. Classical nonnegative matrix factorization

The nonnegative matrix factorization problem can be stated as follows: given a matrix $M \in \mathbb{R}_+^{m_1 \times m_2}$, find a decomposition $M = VH$ with $V \in \mathbb{R}_+^{m_1 \times a}$ and $H \in \mathbb{R}_+^{a \times m_2}$, and with a as small as possible. The minimal inner dimension a for which a decomposition exists is called the positive rank (p-rank) of M. There exist matrices with only trivial minimal decompositions $M = IM$ and $M = MI$. In [11] these matrices are called prime. It is clear that $0 \leqslant \operatorname{rank}(V) \leqslant \text{p-rank}(V) \leqslant \min\{m_1, m_2\}$. There exists a finite algorithm to compute the minimal inner dimension of an exact decomposition. However, the complexity bounds on the number of arithmetic/boolean operations that this algorithm requires, are non-polynomial. Recently, the approximate nonnegative matrix factorization problem was introduced in [8]. The idea is that one chooses the inner dimension a and looks for matrices V and H such that VH approximates P optimally in a certain distance measure. The Kullback–Leibler divergence is a popular such measure. The Kullback–Leibler divergence between two nonnegative matrices of the same size is defined as

$$D(A\|B) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right).$$

The nonnegative matrix factorization problem can now be stated as

Problem 1. Given $M \in \mathbb{R}_+^{m_1\times m_2}$ and given a, minimize $D(M\|VH)$ with respect to V (of size $m_1 \times a$) and H (of size $a \times m_2$), subject to the constraints $V, H \geqslant 0$.

Lee and Sueng propose iterative update formulas to solve Problem 1 and prove interesting properties of the formulas.

Theorem 1 [7,8]. *The divergence $D(M\|VH)$ is nonincreasing under the update rules*

$$H_{i,l} \leftarrow H_{i,l}\frac{\sum_\mu V_{\mu i}\frac{M_{\mu l}}{(VH)_{\mu l}}}{\sum_\mu V_{\mu i}}, \qquad V_{k,i} \leftarrow V_{k,i}\frac{\sum_\nu H_{i\nu}\frac{M_{k\nu}}{(VH)_{k\nu}}}{\sum_\nu H_{i\nu}}.$$

The divergence is invariant under these updates if and only if V and H are in a stationary point of the divergence.

The theorem says that fixed points of the update formulas are stationary points of the cost function $D(M\|VH)$. However, the theorem does not imply convergence of the update formulas. As the initial values for V and H have to be chosen nonnegative, the obtained matrices V and H are nonnegative.

3. Structured nonnegative matrix factorization

The *structured nonnegative matrix factorization* studied in this paper may be stated as follows: given a square matrix $P \in \mathbb{R}_+^{p\times p}$, find a decomposition $P = VAV^\top$ with $V \in \mathbb{R}_+^{p\times a}$, $A \in \mathbb{R}_+^{a\times a}$, and with a as small as possible. We define the structured positive rank (sp-rank) of a square matrix P, to be the minimal dimension a for which a decomposition $P = VAV^\top$ exists. It is intuitively clear that $0 \leqslant \operatorname{rank}(P) \leqslant \text{p-rank}(P) \leqslant \text{sp-rank}(P) \leqslant p$. Again, it can be shown that there exists a finite algorithm to compute the minimal inner dimension of an exact decomposition. However, the complexity bounds on the number of arithmetic/boolean operations that this algorithm requires, are non-polynomial.

Problem 2. Given $P \in \mathbb{R}_+^{p\times p}$ and given a, minimize $D(P\|VAV^\top)$ with respect to V (of size $p \times a$) and A (of size $a \times a$), subject to the constraints $V, A \geqslant 0$.

The partial derivatives of $F(A, V) = D(P\|VAV^\top)$ with respect to the elements $A_{i,j}$ and $V_{k,i}$ of the matrices A and V can be calculated as

$$\frac{\partial F}{\partial A_{ij}}(A, V) = -\sum_{\mu\nu} V_{\mu i}V_{\nu j}\frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} + \sum_{\mu\nu} V_{\mu i}V_{\nu j}, \tag{1}$$

$$\frac{\partial F}{\partial V_{ki}}(A, V) = -\sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda}\frac{P_{k\nu}}{(VAV^\top)_{k\nu}} - \sum_{\lambda\nu} V_{\nu\lambda}A_{\lambda i}\frac{P_{\nu k}}{(VAV^\top)_{\nu k}} + \sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda} + \sum_{\lambda\nu} V_{\nu\lambda}A_{\lambda i}. \tag{2}$$

The Karush–Kuhn–Tucker optimality conditions are

$$V_{k,i} \geqslant 0, \quad A_{i,j} \geqslant 0, \tag{3}$$

$$\frac{\partial F}{\partial A_{ij}}(A, V) \geqslant 0, \quad \frac{\partial F}{\partial V_{ki}}(A, V) \geqslant 0, \tag{4}$$

$$A_{ij}\frac{\partial F}{\partial A_{ij}}(A, V) = 0, \quad V_{ki}\frac{\partial F}{\partial V_{ki}}(A, V) = 0 \tag{5}$$

for $i = 1, 2, \ldots, a$, $j = 1, 2, \ldots, a$, and $k = 1, 2, \ldots, p$.

Now following the approach of [6], we obtain the following theorem.

Theorem 2. *Let $P \in \mathbb{R}^{p\times p}$. Every stationary point (A, V) of the cost function $D(P\|VAV^\top)$ preserves the mean of the row and column sum of P, i.e.*

$$\frac{\sum_l P_{kl} + P_{lk}}{2} = \frac{\sum_l (VAV^\top)_{kl} + (VAV^\top)_{lk}}{2}, \quad k = 1, 2, \ldots, p. \tag{6}$$

As a consequence the element sum of P is also preserved, i.e.

$$\sum_{kl} P_{kl} = \sum_{kl} (VAV^\top)_{kl}. \tag{7}$$

Proof. Eq. (5) gives, for $k = 1, 2, \ldots, p$ and $i = 1, 2, \ldots, a$

$$V_{ki}\frac{\partial F}{\partial V_{ki}}(A, V) = 0.$$

From Eq. (2), we obtain

$$V_{ki}\left(\sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda}\frac{P_{k\nu}}{(VAV^\top)_{k\nu}} + V_{\nu\lambda}A_{\lambda i}\frac{P_{\nu k}}{(VAV^\top)_{\nu k}}\right) = V_{ki}\left(\sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda} + V_{\nu\lambda}A_{\lambda i}\right). \tag{8}$$

The sum over i of the left-hand side of Eq. (8) gives

$$\begin{aligned}
&\sum_i V_{ki}\left(\sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda}\frac{P_{k\nu}}{(VAV^\top)_{k\nu}} + V_{\nu\lambda}A_{\lambda i}\frac{P_{\nu k}}{(VAV^\top)_{\nu k}}\right)\\
&\quad = \sum_\nu (VAV^\top)_{k\nu}\frac{P_{k\nu}}{(VAV^\top)_{k\nu}} + (VAV^\top)_{\nu k}\frac{P_{\nu k}}{(VAV^\top)_{\nu k}}\\
&\quad = \sum_\nu P_{k\nu} + P_{\nu k}.
\end{aligned}$$

On the other hand the sum over i of the right-hand side of Eq. (8) gives

$$\sum_i V_{ki}\left(\sum_{\lambda\nu} V_{\nu\lambda}A_{i\lambda} + V_{\nu\lambda}A_{\lambda i}\right) = \sum_\nu (VAV^\top)_{k\nu} + (VAV^\top)_{\nu k}.$$

This proves the first part of the theorem. The second part of the theorem follows by summing the left- and right-hand side of (6) over k. □

A point $(\widetilde{A}, \widetilde{V})$ is called *normalized* if the matrix $\widetilde{V}$ is column stochastic and $\widetilde{A}$ has the same element sum as P, i.e. $\sum_{ij} \widetilde{A}_{ij} = \sum_{kl} P_{kl}$. As a consequence of Theorem 2, every stationary point (A, V) of the divergence can be written in an equivalent normalized form $(\widetilde{A}, \widetilde{V})$, such that $VAV^\top = \widetilde{V}\widetilde{A}\widetilde{V}^\top$. The matrices $\widetilde{A}$ and $\widetilde{V}$ are given as function of A and V by

$$\widetilde{V} = V(\mathrm{diag}(e^\top V))^{-1},$$
$$\widetilde{A} = (\mathrm{diag}(e^\top V))A(\mathrm{diag}(e^\top V)).$$

The fact that the matrices P and $\widetilde{A}$ have the same element sum follows from $e^\top Pe = e^\top \widetilde{V}\widetilde{A}\widetilde{V}^\top e = e^\top \widetilde{A}e$.

So, for a stationary point (A, V) of the cost function $D(P\|VAV^\top)$, there exists a *normalized* version $(\widetilde{V}, \widetilde{A})$ which gives the same approximation (and hence the same cost function value). Therefore, when minimizing $D(P\|VAV^\top)$ over nonnegative V and A, it suffices to minimize over nonnegative matrices V and A that are normalized. Minimizing over normalized matrices can be done by choosing normalized initial values and by making sure that the update formulas retain the normalization. Choosing normalized initial values is no problem, and the fact that the proposed update formulas retain the normalization, is shown in the proof of Theorem 3. From now on, we assume that V or A are normalized, without explicitly indicating it with a tilde.

Theorem 3. *Assume that the starting values V^0 and A^0 are normalized, i.e. $\sum_{ij} A^0_{ij} = \sum_{kl} P_{kl}$ and $\sum_k V^0_{k,i} = 1, i = 1, 2, \ldots, a$. Then the divergence $D(P\|VAV^\top)$ is nonincreasing under the update rules*

$$A_{ij} \leftarrow A_{ij} \sum_{\mu\nu} V_{\mu i} V_{\nu j} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}}, \tag{9}$$

$$V_{ki} \leftarrow V_{ki} \frac{\sum_{\lambda\nu} \frac{P_{k\nu}}{(VAV^\top)_{k\nu}} A_{i\lambda} V_{\nu\lambda} + \frac{P_{\nu k}}{(VAV^\top)_{\nu k}} A_{\lambda i} V_{\nu\lambda}}{\sum_{\lambda\mu\nu} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} A_{i\lambda} V_{\nu\lambda} V_{\mu i} + \frac{P_{\nu\mu}}{(VAV^\top)_{\nu\mu}} A_{\lambda i} V_{\nu\lambda} V_{\mu i}}. \tag{10}$$

Proof. First note that the update rule for A retains the normalization, since

$$\sum_{ij} A_{ij} \sum_{\mu\nu} V_{\mu i} V_{\nu j} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} = \sum_{\mu\nu} (VAV^\top)_{\mu\nu} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} = \sum_{\mu\nu} P_{\mu\nu}.$$

Also the update rule for V retains the normalization, since

$$\sum_k V_{ki} \frac{\sum_{\lambda\nu} \frac{P_{k\nu}}{(VAV^\top)_{k\nu}} A_{i\lambda} V_{\nu\lambda} + \frac{P_{\nu k}}{(VAV^\top)_{\nu k}} A_{\lambda i} V_{\nu\lambda}}{\sum_{\lambda\mu\nu} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} A_{i\lambda} V_{\nu\lambda} V_{\mu i} + \frac{P_{\nu\mu}}{(VAV^\top)_{\nu\mu}} A_{\lambda i} V_{\nu\lambda} V_{\mu i}} = 1, \quad i = 1, 2, \ldots, a.$$

Next, we prove that the divergence $D(P\|VAV^\top)$ is nonincreasing under an update for A. Note therefore that the cost function $F_A(A)$

$$F_A(A) = D(P\|VAV^\top) = \sum_{\mu\nu} P_{\mu\nu} \log P_{\mu\nu} - P_{\mu\nu} + (VAV^\top)_{\mu\nu} - P_{\mu\nu} \log(VAV^\top)_{\mu\nu}$$

can be approximated by the auxiliary function $G_A(A, A^t)$

$$G_A(A, A^t) = \sum_{\mu\nu} P_{\mu\nu} \log P_{\mu\nu} - P_{\mu\nu} + (VAV^\top)_{\mu\nu} - \sum_{\kappa\lambda} P_{\mu\nu} \frac{V_{\mu\kappa} A^t_{\kappa\lambda} V_{\nu\lambda}}{(VA^tV^\top)_{\mu\nu}} \left(\log V_{\mu\kappa} A_{\kappa\lambda} V_{\nu\lambda} - \log \frac{V_{\mu\kappa} A^t_{\kappa\lambda} V_{\nu\lambda}}{(VA^tV^\top)_{\mu\nu}} \right).$$

Convexity of the $-\log$ function and the fact that $\sum_{\kappa\lambda} \frac{V_{\mu\kappa} A^t_{\kappa\lambda} V_{\nu\lambda}}{(VA^tV^\top)_{\mu\nu}} = 1$ gives $G_A(A, A^t) \geqslant F_A(A)$. In addition $G_A(A^t, A^t) = F_A(A^t)$, and therefore

$$F_A(A^{t+1}) \leqslant G_A(A^{t+1}, A^t) = \min_A G_A(A, A^t) \leqslant G_A(A^t, A^t) = F_A(A^t).$$

To obtain an update formula, we put A^{t+1} equal to the minimizer of $G_A(A, A^t)$. From

$$\frac{\partial G_A}{\partial A_{ij}}(A, A^t) = \sum_{\mu\nu} V_{\mu i} V_{\nu k} - \sum_{\mu\nu} P_{\mu\nu} \frac{V_{\mu i} A^t_{ij} V_{\nu j}}{(VA^tV^\top)_{\mu\nu}} \frac{1}{A_{ij}} = 0,$$

we obtain

$$A^{t+1}_{ij} = A^t_{ij} \frac{\sum_{\mu\nu} V_{\mu i} V_{\nu j} \frac{P_{\mu\nu}}{(VA^tV^\top)_{\mu\nu}}}{\sum_{\mu\nu} V_{\mu i} V_{\nu k}}. \tag{11}$$

Since the denominator is equal to $(\sum_\mu V_{\mu i})(\sum_\nu V_{\nu k}) = 1$, this yields the proposed update formula (9).

We now prove that the divergence $D(P \| VAV^\top)$ is also nonincreasing under an update for V. In order to see this, note that the cost function $F_V(V)$

$$F_V(V) = D(P \| VAV^\top) = \sum_{\mu\nu} P_{\mu\nu} \log P_{\mu\nu} - P_{\mu\nu} + (VAV^\top)_{\mu\nu} - P_{\mu\nu} \log (VAV^\top)_{\mu\nu}$$

can be approximated by the auxiliary function $G_V(V, V^t)$

$$G_V(V, V^t) = \sum_{\mu\nu} P_{\mu\nu} \log P_{\mu\nu} - P_{\mu\nu} + (V\overline{A}(V^t)^\top)_{\mu\nu} + (V^t\overline{A}V^\top)_{\mu\nu} - (V^t\overline{A}(V^t)^\top)_{\mu\nu} - \sum_{\kappa\lambda} P_{\mu\nu} \frac{V^t_{\mu\kappa} A_{\kappa\lambda} V^t_{\nu\lambda}}{(V^tA(V^t)^\top)_{\mu\nu}} \left(\log V_{\mu\kappa} A_{\kappa\lambda} V_{\nu\lambda} - \log \frac{V^t_{\mu\kappa} A_{\kappa\lambda} V^t_{\nu\lambda}}{(V^tA(V^t)^\top)_{\mu\nu}} \right),$$

where $\overline{A}$ will be chosen later on. At this moment it suffices to require that $\sum_{ij} \overline{A}_{ij} = \sum_{kl} P_{kl}$. From the fact that $\sum_k V_{ki} = \sum_k V^t_{ki} = 1$ and $\sum_{ij} A_{ij} = \sum_{ij} \overline{A}_{ij} = \sum_{kl} P_{kl}$, we have that

$$\sum_{\mu\nu} (VAV^\top)_{\mu\nu} = \sum_{\mu\nu} (V\overline{A}(V^t)^\top)_{\mu\nu} + (V^t\overline{A}V^\top)_{\mu\nu} - (V^t\overline{A}(V^t)^\top)_{\mu\nu}.$$

From this, the convexity of the $-\log$ function, and the fact that $\sum_{\kappa\lambda} \frac{V^t_{\mu\kappa} A_{\kappa\lambda} V^t_{\nu\lambda}}{(V^tA(V^t)^\top)_{\mu\nu}} = 1$, we obtain that $G_V(V, V^t) \geqslant F_V(V)$. In addition, since $G_V(V^t, V^t) = F_V(V^t)$ there holds

$$F_V(V^{t+1}) \leqslant G_V(V^{t+1}, V^t) = \min_V G_V(V, V^t) \leqslant G_V(V^t, V^t) = F_V(V^t).$$

So to obtain an update formula, we put V^{t+1} equal to the minimizer of $G_V(V, V^t)$. From

$$\frac{\partial G_V}{\partial V_{ki}}(V, V^t) = \sum_{\lambda\nu} \overline{A}_{i\lambda} V^t_{\nu\lambda} + \overline{A}_{\lambda i} V^t_{\nu\lambda} - P_{k\nu} \frac{V^t_{ki} A_{i\lambda} V^t_{\nu\lambda}}{(V^t A (V^t)^\top)_{k\nu}} \frac{1}{V_{ki}} - P_{\nu k} \frac{V^t_{\nu\lambda} A_{\lambda i} V^t_{ki}}{(V^t A (V^t)^\top)_{\nu k}} \frac{1}{V_{ki}} = 0,$$

we find that

$$V^{t+1}_{ki} = V^t_{ki} \frac{\sum_{\nu\lambda} A_{i\lambda} V^t_{\nu\lambda} \frac{P_{k\nu}}{(V^t A (V^t)^\top)_{k\nu}} + A_{\lambda i} V^t_{\nu\lambda} \frac{P_{\nu k}}{(V^t A (V^t)^\top)_{\nu k}}}{\sum_{\lambda\nu} \overline{A}_{i\lambda} V^t_{\nu\lambda} + \overline{A}_{\lambda i} V^t_{\nu\lambda}}. \tag{12}$$

One can easily see that the denominator is equal to $\sum_\lambda \overline{A}_{i\lambda} + \overline{A}_{\lambda i}$. By taking

$$\overline{A}_{ij} = A_{ij} \sum_{\mu\nu} V^t_{\mu i} V^t_{\nu j} \frac{P_{\mu\nu}}{(V^t A (V^t)^\top)_{\mu\nu}}, \tag{13}$$

we obtain the proposed update formula for V. □

We have proven that the divergence $D(P\|VAV^\top)$ is nonincreasing under the update rules (9) and (10). We now consider the invariant points of the update formulas and investigate their relation with the stationary points of the divergence $D(P\|VAV^\top)$.

For fixed $V^t = V^0$, the divergence $F(A, V) = D(P\|VAV^\top)$ is invariant under an update of A, i.e. $A^{t+1} = A^t$, if and only if A^t is a stationary point of the divergence with fixed V^0, i.e. $A^t_{ij} \frac{\partial F}{\partial A_{ij}}(A^t, V^0) = 0$. For fixed $A^t = A^0$ on the other hand, the divergence is invariant under an update for V, i.e. $V^{t+1} = V^t$, if and only if

$$V^t_{ki} \left(\sum_{\nu\lambda} \overline{A}^0_{i\lambda} + \overline{A}^0_{\lambda i} - A^0_{i\lambda} V^t_{\nu\lambda} \frac{P_{k\nu}}{(V^t A^0 (V^t)^\top)_{k\nu}} - A^0_{\lambda i} V^t_{\nu\lambda} \frac{P_{\nu k}}{(V^t A^0 (V^t)^\top)_{\nu k}} \right) = 0.$$

Notice that this last condition is in general not equivalent to the condition that V^t is a stationary point of the divergence with fixed A^0. So for the case where we take A fixed and update only V, it is possible that the formulas, if they converge, converge to a point that is not a stationary point of the divergence with fixed A^0.

However, if we use the update formulas for A and V alternatingly, i.e.

$$(A^0, V^0) \mapsto (A^1, V^0) \mapsto (A^1, V^1) \mapsto (A^2, V^1) \mapsto (A^2, V^2) \mapsto \cdots,$$

we have the following result.

Theorem 4. *The divergence is invariant under updates* (9) *and* (10) *if and only if* (A, V) *is a stationary point of the divergence, i.e.*

$$\begin{cases} A^{t+1} = A^t, \\ V^{t+1} = V^t, \end{cases} \quad \Leftrightarrow \quad \begin{cases} A^t_{ij} \frac{\partial F}{\partial A_{ij}}(A^t, V^t) = 0, & i = 1, 2, \ldots a; \quad j = 1, 2, \ldots, a, \\ V^t_{ki} \frac{\partial F}{\partial V_{ki}}(A^t, V^t) = 0, & k = 1, 2, \ldots p; \quad i = 1, 2, \ldots, a. \end{cases}$$

Proof. We first prove the $\Leftarrow$ part. From the fact that $A^t_{ij} \frac{\partial F}{\partial A_{ij}}(A^t, V^t) = 0$ for $i = 1, 2, \ldots, a$; $j = 1, 2, \ldots, a$, it follows that for a certain i, j either $A^t_{ij} = 0$ or $\frac{\partial F}{\partial A_{ij}}(A^t, V^t) = 0$. In the first case,

the updated value A_{ij}^{t+1} is also equal to 0 because the update is multiplicative. In the second case, the update factor for A_{ij} is equal to 1. So, in both cases, we have $A_{ij}^{t+1} = A_{ij}^t$. It also follows that $A_{ij}^{t+1} \frac{\partial F}{\partial A_{ij}}(A^{t+1}, V^t) = 0$, from which we conclude that $\overline{A}_{ij}^{t+1} = A_{ij}^{t+1}$. Since $V_{ki}^t \frac{\partial F}{\partial V_{ki}}(A^t, V^t) = 0$ for $i = 1, 2, \ldots, a$; $j = 1, 2, \ldots, a$, we either have $V_{ki}^t = 0$, or $\frac{\partial F}{\partial V_{ki}}(A^{t+1}, V^t) = 0$. In the first case the updated value V_{ki}^{t+1} is also equal to 0. In the second case, one can see from $\overline{A}_{ij}^{t+1} = A_{ij}^{t+1}$ and $\frac{\partial F}{\partial V_{ki}}(A^{t+1}, V^t) = 0$, that the update factor for V_{ki} is equal to 1. So in both cases we have that $V_{ki}^{t+1} = V_{ki}^t$. This proves the first part of the theorem.

Next, we prove the $\Rightarrow$ part. From the fact that $A^{t+1} = A^t$, we conclude that either $A_{ij}^t = 0$ or the update factor for A_{ij} is equal to 1. This implies that $A_{ij}^t \frac{\partial F}{\partial A_{ij}}(A^t, V^t) = 0$. From $A_{ij}^{t+1} \frac{\partial F}{\partial A_{ij}}(A^{t+1}, V^t) = 0$, we obtain $\overline{A}_{ij}^{t+1} = A_{ij}^{t+1}$. From this and $V^{t+1} = V^t$, we conclude that $V_{ki}^t \frac{\partial F}{\partial V_{ki}}(A^{t+1}, V^t) = 0$. This proves the second part of the theorem. □

It follows that if the update formulas converge, that they converge to a stationary point of the cost function in case A and V are updated alternatingly or in case V is fixed and A is updated. However, when A is fixed and only V is updated, it is only guarantueed that the divergence is nonincreasing. It is possible that the formulas converge to a point that is not a stationary point of the divergence with fixed A.

The scheme below implements the same update formulas as in Theorem 3, but is much better from computational point of view:

$$A_{ij} \leftarrow A_{ij} \sum_{\mu\nu} V_{\mu i} V_{\nu j} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}},$$

$$V_{ki} \leftarrow V_{ki} \sum_{\lambda\nu} \frac{P_{k\nu}}{(VAV^\top)_{k\nu}} A_{i\lambda} V_{\nu\lambda} + \frac{P_{\nu k}}{(VAV^\top)_{\nu k}} A_{\lambda i} V_{\nu\lambda},$$

normalize V such that $e^\top V = e^\top$.

In case P is symmetric and $VAV^\top$ is an approximation of P then one can easily see that $\frac{VAV^\top + (VAV^\top)^\top}{2}$ is a better (or equally good) approximation of P. So if the matrix P is symmetric, we can restrict our search to symmetric approximations. On the other hand, every symmetric approximation $VAV^\top$ can be transformed to a form where A is symmetric by taking $V(\frac{A+A^\top}{2})V^\top$. So in case P is symmetric, we can restrict our search to approximations with $A = A^\top$. This restriction is easy to fulfill in practice since the update formula for A (formula (9)) retains the symmetry. So by starting with a symmetric $A^0 = (A^0)^\top$, we end up with a symmetric A.

4. Symmetric nonnegative matrix factorization

Another decomposition of interest is the *symmetric nonnegative matrix factorization*. In this problem, we are given a square, symmetric, nonnegative definite matrix $P \in \mathbb{R}_+^{p\times p}$, and are looking for a decomposition $P = VV^\top$ with $V \in \mathbb{R}_+^{p\times a}$. The completely positive rank (cp-rank) [1] of the matrix P is the minimal inner dimension for which a decomposition $P = VV^\top$ exists. In contrast to the nonnegative matrix factorization and the structured nonnegative matrix factorization, a symmetric nonnegative factorization with a finite inner dimension does not always exist. In case the symmetric decomposition does not exist, we say that the completely positive rank is infinite. Obviously $0 \leqslant \text{rank}(P) \leqslant \text{p-rank}(P) \leqslant \text{sp-rank}(P) \leqslant \text{cp-rank}(P)$. We consider the following approximate problem.

Problem 3. Given a symmetric matrix $P \in \mathbb{R}_+^{p\times p}$ and given a, minimize $D(P\|VV^\top)$ with respect to V (of size $p \times a$), subject to the constraint $V \geqslant 0$.

Analogous to the decomposition $VAV^\top$, one can prove that the row (or column) sum of P is equal to the row (or column) sum of $VV^\top$, where V is a stationary point of the divergence $D(P|VV^\top)$. As a consequence the element sum of P is equal to the element sum of $VV^\top$ with V a stationary point of the divergence.

Theorem 5. *Given a nonnegative matrix $P \in \mathbb{R}^{p\times p}$, then every stationary point V of the cost function $D(P\|VV^\top)$ preserves the element sum of P, i.e.*

$$\sum_{kl} P_{kl} = \sum_{kl} (VV^\top)_{kl}.$$

As a consequence of this theorem, we see that for every stationary point V of the divergence $D(P\|VV^\top)$, there exists a matrix $\widetilde{V}$ and a diagonal matrix D such that $VV^\top = \widetilde{V}D\widetilde{V}^\top$, where $\widetilde{V}$ is row stochastic and the element sum of P equals the element sum of D, i.e. $\sum_{kl} P_{kl} = \sum_i D_{ii}$.

Our approach for Problem 3 is to look for a decomposition $P \simeq \widetilde{V}D\widetilde{V}^\top$, with $\widetilde{V}$ column stochastic and D diagonal with sum of its elements equal to the element sum of P such that the divergence $F(D, \widetilde{V}) = D(P\|\widetilde{V}D\widetilde{V}^\top)$ is minimized. This leads to updates which make the divergence decrease, and are invariant if and only if we have reached a stationary point of the divergence $D(P\|\widetilde{V}D\widetilde{V}^\top)$. Once an approximation $P \simeq \widetilde{V}D\widetilde{V}^\top$ has been found, we obtain a decomposition of the form $VV^\top$, by calculating V as

$$V = \widetilde{V}\sqrt{D}.$$

The next theorem proposes update formulas for the decomposition $P = \widetilde{V}D\widetilde{V}^\top$.

Theorem 6. *Under the condition that the starting values $\widetilde{V}^0$ and D^0 are normalized, i.e. $\sum_i D^0_{i,i} = \sum_{k,l} P_{k,l}$ and $\sum_k \widetilde{V}^0_{k,i} = 1, i = 1, 2, \ldots, a$, the divergence $D(P\|\widetilde{V}D\widetilde{V}^\top)$ is nonincreasing under the update rules*

$$D_{ii} \leftarrow D_{ii} \sum_{\mu\nu} \widetilde{V}_{\mu i}\widetilde{V}_{\nu i} \frac{P_{\mu\nu}}{(\widetilde{V}D\widetilde{V}^\top)_{\mu\nu}}, \tag{14}$$

$$\widetilde{V}_{ki} \leftarrow \widetilde{V}_{ki} \frac{\sum_\nu \frac{P_{k\nu}}{(\widetilde{V}D\widetilde{V}^\top)_{k\nu}} D_{ii}\widetilde{V}_{\nu i} + \frac{P_{\nu k}}{(\widetilde{V}D\widetilde{V}^\top)_{\nu k}} D_{ii}\widetilde{V}_{\nu i}}{\sum_{\mu\nu} \frac{P_{\mu\nu}}{(\widetilde{V}D\widetilde{V}^\top)_{\mu\nu}} D_{ii}\widetilde{V}_{\nu i}\widetilde{V}_{\mu i} + \frac{P_{\nu\mu}}{(\widetilde{V}D\widetilde{V}^\top)_{\nu\mu}} D_{ii}\widetilde{V}_{\nu i}\widetilde{V}_{\mu i}}. \tag{15}$$

Proof. The proof is analogous to the proof of Theorem 3. □

If we use the update formulas for D and $\widetilde{V}$ alternatingly, i.e.

$$(D^0, \widetilde{V}^0) \mapsto (D^1, \widetilde{V}^0) \mapsto (D^1, \widetilde{V}^1) \mapsto (D^2, \widetilde{V}^1) \mapsto (D^2, \widetilde{V}^2) \mapsto \cdots,$$

we can prove the following theorem.

Theorem 7. *The divergence is invariant under updates (14) and (15) if and only if $(D, \widetilde{V})$ is a stationary point of the divergence, i.e.*

$$\begin{cases} D^{t+1} = D^t, \\ \widetilde{V}^{t+1} = \widetilde{V}^t, \end{cases} \quad \Leftrightarrow \quad \begin{cases} D^t_{ii} \frac{\partial F}{\partial D_{ii}}(D^t, \widetilde{V}^t) = 0, & i = 1, 2, \ldots, a, \\ \widetilde{V}^t_{ki} \frac{\partial F}{\partial \widetilde{V}_{ki}}(D^t, \widetilde{V}^t) = 0, & k = 1, 2, \ldots, p; \;\; i = 1, 2, \ldots, a. \end{cases}$$

Proof. The proof is analogous to the proof of Theorem 4. □

5. Application to hidden Markov realization theory

Hidden Markov models (HMMs) are used as a modeling tool for finite valued stochastic processes. They are used in speech processing, image processing, bioinformatics, etc. Although HMMs were introduced in literature in 1950s, many theoretical questions remain open until now. One of these is the realization problem, i.e. given string probabilities, find the system matrices of the underlying HMM. In [4], it is shown that the classical nonnegative matrix factorization can be used for realization of HMMs. In this section, we show that the (approximate) HMM realization problem for string probabilities up to length two can be solved using the matrix decomposition techniques of this paper.

An hidden Markov model [10] consists of a finite valued stochastic state process x and a finite valued output process y that depends in a probabilitic manner of the state process. An hidden Markov model is completely defined by $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, B, \pi(1))$, where

- $\mathbb{X} = \{1, 2, \ldots, |\mathbb{X}|\}$ with $|\mathbb{X}| < \infty$ is the state alphabet, and $\mathbb{Y}$ with $|\mathbb{Y}| < \infty$ is the output alphabet;
- $\pi(1)$ is a row vector in $\mathbb{R}_+^{|\mathbb{X}|}$ with $\pi(1)e = 1$ and $\pi_i(1) = P(x(1) = i)$, the probability distribution of the initial state;
- $\Pi_{\mathbb{X}}$ is a matrix in $\mathbb{R}_+^{|\mathbb{X}|\times|\mathbb{X}|}$ with $\Pi_{\mathbb{X}} e = e$ and $(\Pi_{\mathbb{X}})_{ij} = P(x(t+1) = j | x(t) = i)$, the probability of going from state i to state j;
- B is a matrix in $\mathbb{R}_+^{|\mathbb{X}|\times|\mathbb{Y}|}$ with $Be = e$ and $B_{ik} = P(y(t) = \mathrm{y}_k | x(t) = i)$, the probability of producing output symbol k given that the present state is i, where $(\mathrm{y}_k, k = 1, 2, \ldots, |\mathbb{Y}|)$ is an ordering of the symbols of the set $\mathbb{Y}$;

Define $\mathbb{Y}^*$ as the set of strings of finite length with symbols from $\mathbb{Y}$. String probabilities $\mathscr{P}$: $\mathbb{Y}^* \mapsto [0, 1]$ are then defined as

$$\mathscr{P}(\mathbf{u}) := P(y(1) = u_1, y(2) = u_2, \ldots, y(|\mathbf{u}|) = u_{|\mathbf{u}|}),$$

where $\mathbf{u} = u_1 u_2 \cdots u_{|\mathbf{u}|} \in \mathbb{Y}^*$. The matrix P is defined as the $|\mathbb{Y}| \times |\mathbb{Y}|$ matrix with k, lth element $\mathscr{P}(\mathrm{y}_k \mathrm{y}_l)$, where $\mathrm{y}_k \mathrm{y}_l$ is the concatenation of y_k and y_l. Notice that P contains all string probabilities of strings of length 2. The element sum of P is equal to 1, i.e. $\sum_{kl} P_{kl} = 1$.

It can be shown that P containing string probabilities of a hidden Markov model satisfy

$$P = B^\top \mathrm{diag}(\pi(1)) \Pi_{\mathbb{X}} B. \tag{16}$$

In the HMM realization problem, we are given the string probabilities of all possible finite strings and the problem is to find a HMM with minimal state dimension that realizes $\mathscr{P}$, i.e. to find a HMM that produces exactly the given string probabilities. In this paper, we consider the HMM realization problem for string probabilities of length two.

Because of (16), the problem of finding a HMM with given probabilities of strings of length two, is equivalent to the problem of finding, for a given $P \in \mathbb{R}_+^{|\mathbb{Y}|\times|\mathbb{Y}|}$ with $e^\top P e = 1$, matrices

$B \in \mathbb{R}_+^{|\mathbb{X}|\times|\mathbb{Y}|}$, with $Be = e$, and $\Pi_{\mathbb{X}} \in \mathbb{R}_+^{|\mathbb{X}|\times|\mathbb{X}|}$, with $\Pi_{\mathbb{X}} e = e$, and a vector $\pi(1) \in \mathbb{R}_+^{|\mathbb{X}|}$ with $\pi(1)e = 1$, such that $P = B^\top \mathrm{diag}(\pi(1))\Pi_{\mathbb{X}} B$, where $|\mathbb{X}|$ is as small as possible.

In the previous sections of this paper, we developed a method to (approximately) decompose a matrix P into a product $VAV^\top$, with V and A positive, V column stochastic and with the element sum of A equal to the element sum of P. It is easy to see that this method allows us to solve the realization problem approximately. Indeed, from an approximate decomposition $P \simeq VAV^\top$, one can find B, $\Pi_{\mathbb{X}}$ and $\pi(1)$ as follows:

$$
\begin{aligned}
B &= V^\top, \\
\Pi_{\mathbb{X}} &= (\mathrm{diag}(Ae))^{-1} A, \\
\pi(1) &= (Ae)^\top.
\end{aligned}
$$

This defines a hidden Markov realization which approximately models string probabilities up to length two.

Consider the following simulation example. Suppose that we are given the probabilities of strings of length two for an HMM with $\mathbb{Y} = \{a, b, \ldots, j\}$, and that we want to find the system matrices of the underlying HMM. Suppose there is an ordering $(\mathrm{y}_1 = a, \mathrm{y}_2 = b, \ldots, \mathrm{y}_{10} = j)$ on the output set $\mathbb{Y}$. Now the string probabilities can be stacked in the matrix P as described before, i.e. $P_{kl} = \mathscr{P}(\mathrm{y}_k \mathrm{y}_l)$.

In our simulation example P is given by

$$
P = \begin{bmatrix}
396 & 193 & 149 & 116 & 113 & 94 & 98 & 161 & 128 & 454 \\
182 & 128 & 87 & 85 & 77 & 67 & 70 & 120 & 84 & 191 \\
150 & 87 & 69 & 60 & 58 & 52 & 53 & 77 & 63 & 150 \\
111 & 84 & 60 & 61 & 55 & 51 & 52 & 80 & 57 & 112 \\
112 & 75 & 58 & 55 & 51 & 47 & 48 & 70 & 54 & 105 \\
92 & 67 & 50 & 51 & 46 & 45 & 45 & 63 & 47 & 93 \\
97 & 69 & 52 & 52 & 47 & 46 & 46 & 65 & 49 & 96 \\
149 & 118 & 78 & 80 & 72 & 63 & 65 & 114 & 78 & 148 \\
126 & 81 & 64 & 58 & 55 & 49 & 51 & 75 & 60 & 113 \\
488 & 189 & 152 & 105 & 100 & 86 & 90 & 141 & 111 & 415
\end{bmatrix} 10^{-4}.
$$

This matrix P was generated as $P = B^\top \mathrm{diag}(\pi)\Pi_{\mathbb{X}} B$ where B, $\Pi_{\mathbb{X}}$ and $\pi(1)$ are the system matrices of an HMM $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, B, \pi)$ with $\mathbb{X} = \{1, 2, \ldots, 5\}$ and

$$
\Pi_{\mathbb{X}} = \begin{bmatrix}
0.80 & 0.00 & 0.10 & 0.10 & 0.00 \\
0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\
0.40 & 0.10 & 0.30 & 0.20 & 0.00 \\
0.15 & 0.05 & 0.10 & 0.35 & 0.35 \\
0.05 & 0.05 & 0.05 & 0.55 & 0.30
\end{bmatrix},
$$

$$
\pi = \begin{bmatrix} 0.4850 & 0.0375 & 0.1218 & 0.2300 & 0.1257 \end{bmatrix},
$$

$$
B = \begin{bmatrix}
0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 \\
0.15 & 0.00 & 0.25 & 0.00 & 0.20 & 0.00 & 0.05 & 0.00 & 0.35 & 0.00 \\
0.30 & 0.30 & 0.00 & 0.10 & 0.00 & 0.00 & 0.00 & 0.30 & 0.00 & 0.00 \\
0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.55 \\
0.70 & 0.10 & 0.10 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.10
\end{bmatrix}.
$$

Table 1
Number of iterations for the multiplicative update method minimizing the Kullback–Leibler divergence

sp-Rank	1	2	3	4	5	6	7	8	9	10
Number of iterations	1	272	1439	1431	2137	3656	2157	2320	1786	1806

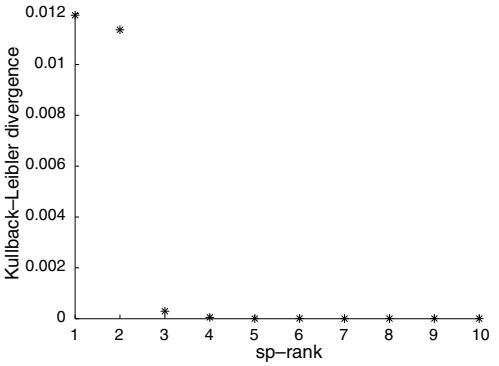

Fig. 1. Kullback–Leibler divergence between the true matrix P and its optimal (w.r.t. the Kullback–Leibler divergence) approximation of sp-rank 1, 2, . . . , 10 computed with the iterative algorithm of Theorem 3.

In the simulation example, this model is unknown, but we give it here to check the performance of the algorithms.

We use the iterative update algorithm of Theorem 3 to compute optimal approximations with respect to the Kullback–Leibler divergence with sp-rank equal to 1, 2, . . . , 10. As initial values for the iterative algorithm we use randomly chosen nonnegative matrices. As stopping rule, we use the Kullback–Leibler divergence between the approximation at iteration step t and the approximation at step $t + 1$. The algorithm stops if this divergence is smaller than 10^{-8}. In Table 1, we show the number of steps until convergence for the different sp-ranks.

In Fig. 1, we plot the Kullback–Leibler divergence between the original matrix P and its optimal approximation with respect to the Kullback–Leibler divergence as a function of the sp-rank.

Table 2
String probabilities for strings of length 2

Sequence	Exact	Order 5	Order 4	Order 3	Order 2	Order 1
aa	0.0396	0.0397	0.0396	0.0397	0.0333	0.0362
ab	0.0193	0.0192	0.0192	0.0190	0.0204	0.0207
ac	0.0149	0.0149	0.0149	0.0150	0.0153	0.0156
ad	0.0116	0.0116	0.0116	0.0116	0.0137	0.0137
ae	0.0113	0.0113	0.0113	0.0114	0.0131	0.0128
af	0.0094	0.0094	0.0094	0.0095	0.0113	0.0114
ag	0.0098	0.0098	0.0099	0.0100	0.0118	0.0118
ah	0.0161	0.0161	0.0161	0.0158	0.0185	0.0184
ai	0.0128	0.0128	0.0127	0.0127	0.0144	0.0139
aj	0.0454	0.0454	0.0454	0.0454	0.0384	0.0357

Notice that the divergence is almost equal to 0 for sp-ranks 5–10. This makes sense as the matrix P was generated using an underlying hidden Markov model of order 5. To show further the quality of the approximations, we give in Table 2 the true output probabilities of a selection of length-2 strings and compare them with the probabilities found with the Kullback–Leibler minimalisation method of order 5, 4, ..., 1. We conclude that the approximate HMM realization problem of string probabilities of strings of length 2 can be solved using the matrix factorization method of this paper.

6. Application to clustering based on distance matrices

In the clustering problem, one is given p points $y_1, y_2, \ldots, y_p$ in $\mathbb{R}^n$ and the objective is to find *clusters* of points which are close to each other according to a certain distance measure $d(\cdot, \cdot)$. In general the number of clusters is not known beforehand. We consider the clustering problem where the distance matrix P between the points is given, i.e. $P_{kl} = d(y_k, y_l)$. Note that P is symmetric and that the diagonal elements of P are equal to 0.

A clustering with a clusters $\{C_1, C_2, \ldots, C_a\}$ is a partition of the set $\{y_1, y_2, \ldots, y_p\}$. A clustering is completely described by the matrix $V \in \{0, 1\}^{p\times a}$ defined as

$$V_{k,i} = \begin{cases} 1, & y_k \in C_i, \\ 0, & y_k \notin C_i. \end{cases}$$

Since every point belongs to exactly one cluster, we have $Ve = e$. Define now the *mean distance* A_{ij} between the clusters C_i and C_j as the mean of the distances between every possible combination of a point of C_i and a point of C_j. It follows that the matrix A with as i, jth element the mean distance between cluster C_i and cluster C_j can be calculated as

$$A = V^\dagger P (V^\dagger)^\top,$$

where $V^\dagger = (\mathrm{diag}(e^\top V))^{-1} V^\top$ is the left inverse of V. Notice that the diagonal elements A_{ii} are equal to the mean distance of the points inside cluster C_i. They are hence not necessarily equal to zero.

As a result of the clustering, the distance P_{kl} between two points y_k and y_l is approximated with the mean distance $\widetilde{P}_{kl}$ between the clusters to which the points y_k and y_l belong. The complete matrix $\widetilde{P}$ can be written as

$$\widetilde{P} = VAV^\top.$$

From all the above, we conclude that the clustering of p points with distance matrix P into a clusters, can be expressed as the following matrix factorization problem:

$$\begin{aligned} \text{minimize} \quad & C(P, VV^\dagger\ P\ V^{\dagger\top} V^\top) \\ \text{subject to} \quad & V \in \{0, 1\}^{p\times a}, \\ & Ve = e, \end{aligned}$$

where $C(X, Y)$ is a distance measure between X and Y. As this problem is hard to solve, we propose the following relaxed version:

$$\begin{aligned} \text{minimize} \quad & D(P \| VAV^\top) \\ \text{subject to} \quad & V \in \mathbb{R}_+^{p\times a}, \quad A = A^\top \in \mathbb{R}_+^{a\times a}. \end{aligned}$$

This problem can be solved with the methods proposed in Section 3. A point y_k is assigned to cluster i if $V_{ki} = \max_t V_{kt}$. By using this relaxed version of the problem, we have two additional

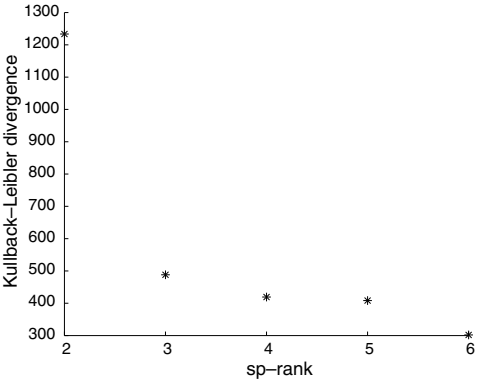

Fig. 2. Kullback–Leibler divergence between the true distance matrix P and its optimal (w.r.t. the Kullback–Leibler divergence) approximation of structured rank $2, \ldots, 6$ computed with the iterative algorithm of Section 3.

advantages. First of all the quality of the clustering can be measured by the ratio between the diagonal and off-diagonal elements of A. The smaller the diagonal elements of A compared to the off-diagonal elements of A, the better the clustering. In addition, one has a measure for how strong a certain point belongs to its cluster. We say that y_k belongs to cluster i if V_{ki} is the biggest element of row $V_{k:}$. If all other elements of the row $V_{k:}$ are much smaller than V_{ki}, one can say that the point y_k strongly belongs to cluster i. If there are elements in the row $V_{k:}$ that are of the same order of magnitude as V_{ki}, we conclude that the point weakly belongs to cluster i.

We now apply this algorithm to a data set of iris flowers (this data set is available in the SOM-toolbox for Matlab as the file `iris.mat`). The data set contains 150 data points $y_1, \ldots, y_{150}$. Each point contains four measurements of an iris flower. The four measurements are the petal width, petal length, sepal width and sepal length of the flower. In the data set three different types of flowers are present, the first 50 samples are Setosa flowers, the next 50 are Versicolor and the last 50 are Virginica flowers. As a first step we make a distance matrix P of size 150×150 with $P_{kl} = \|y_k - y_l\|_2$. Next we decompose the matrix P into a product $VAV^\top$. As we do not know the number of clusters in advance, we make decompositions with inner dimensions 2–6. In Fig. 2, we show the distance between the original matrix P and its approximation $VAV^\top$ as a function of the structured positive rank of the approximation. One sees that the divergence does not decrease much by taking the structured positive rank higher than 3. For that reason, we conclude to work with three clusters.

The A-matrix of the decomposition is given by

$$A = \begin{bmatrix} 0,00000000001 & 4975,78659538 & 9668,80064875 \\ 4975,78659538 & 0,00000032641 & 13768,5062877 \\ 9668,80064875 & 13768,5062877 & 27,0549877873 \end{bmatrix}.$$

Table 3
Clustering result for the points y_{14}, y_{53}, y_{58} and y_{130}

Point y_k	True cluster	$V(k,1)$	$V(k,2)$	$V(k,3)$	Estimated cluster
14	3	0.017	0.012	0.196	3
53	1	0.080	0.115	0.001	2
58	1	0.123	0.000	0.071	1
130	2	0.002	0.201	0.015	2

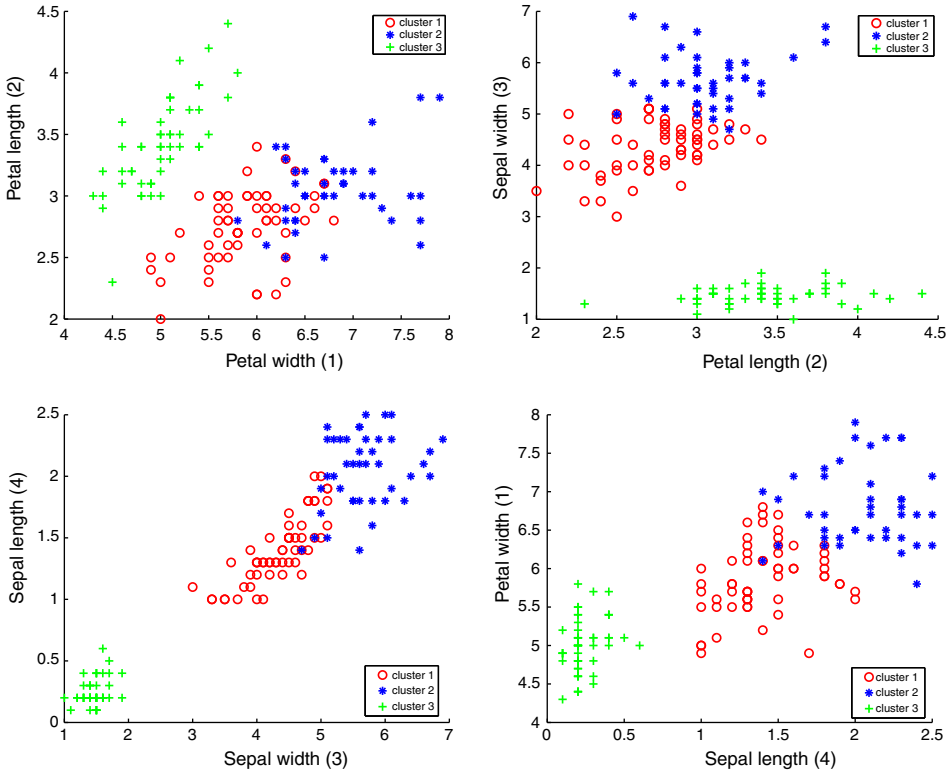

Fig. 3. Visualisation of the result of our clustering algorithm. Points that belong to cluster 1 are plotted with o, points belonging to cluster 2 are plotted with *, and points belonging to cluster 3 are plotted with +.

Notice that the diagonal elements of A are small compared to the off-diagonal elements of A, which is an indication that we have a good clustering. Next to that, the diagonal elements give a measure of the density of the clusters. We conclude that the first cluster is the densest while the third cluster is the least dense. The off-diagonal elements of A give an idea of the mean distance between the clusters.

As explained before, the biggest element of the kth row of the matrix V allow us to conclude to which cluster point k belongs. Using this approach 136 of the 150 iris flowers are clustered correctly (i.e. Versicolor flowers in cluster 1, Virginica flowers in cluster 2 and Setosa flowers in cluster 3). Moreover, the elements of $V_{k:}$ also give a measure of the strength with which the point belong to its cluster. In Table 3, we show the kth row of the matrix V for $k = 14, 53, 58, 130$. For instance for point y_{14} we conclude from our algorithm that it strongly belongs to cluster 3 (as $0.196 \gg 0.017$ and $0.196 \gg 0.012$). This makes sense as the true cluster of that point is indeed cluster 3. On the other hand, point y_{53} (which is one of the miss-clustered points) was connected to cluster 2 but the connection is not strong, its affinity with cluster 1 is almost as high as its affinity with cluster 2. This again makes sense as the true cluster of this point was cluster 1. In Fig. 3, we visualize the result of our clustering algorithm. Points that belong to cluster 1 are plotted with o, point belonging to cluster 2 are plotted with $*$, and points belonging to cluster 3 are plotted with $+$.

7. Conclusion

In this paper we considered the approximate nonnegative matrix factorization $P \simeq VAV^\top$ with the dimension of A small. As distance measure between the matrix P and its approximation we used the Kullback–Leibler divergence. We proved that the element sum of a local optimal approximation $VAV^\top$ equals the element sum of the original matrix P. This result allowed us to work with normalized decompositions with V column stochastic and the element sum of A equal to the element sum of P. We further proved iterative update formulas for V and A that are guarantueed to decrease the Kullback–Leibler divergence between P and $VAV^\top$ and retain the nonnegativity of V and A. As a special case, we commented on the situation where the original matrix P is symmetric. As a final contribution, we proposed iterative update formulas for the approximate decomposition $P \simeq VV^\top$. The decomposition was applied to the hidden Markov realization problem and to the clustering of data points.

Acknowledgements

The authors thank the anonymous referee for helpful comments and suggestions. Bart Vanluyten is a research assistant with the Fund for Scientific Research Flanders (FWO-Vlaanderen). Jan Willems and Bart De Moor are professor with the Katholieke Universiteit Leuven. The SISTA research program is supported by: Research Council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering, several Ph.D./postdoc and fellow Grants; Flemish Government: FWO: Ph.D./postdoc Grants, projects, G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel, research communities (ICCoS, ANMMM, MLDM); IWT: Ph.D. Grants, McKnow-E, Eureka-Flite2; Belgian Federal Science Policy Office: IUAP P6/04 (Dynamical systems, control and optimization, 2007-2011); EU: ERNSI.

References

[1] A. Berman, N. Shaked-Monderer, Completely Positive Matrices, World Scientific Publishing Co., New Jersey, 2003.
[2] M. Berry, M. Browne, Algorithms and applications for approximate nonnegative matrix factorization, submitted for publication.
[3] M. Catral, L. Han, M. Neumann, R. Plemmons, On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices, Linear Algebra Appl. 393 (2004) 107–126.
[4] L. Finesso, A. Grassi, P. Spreij, Approximation of stationary processes by hidden Markov models, submitted for publication.
[5] L. Finesso, P. Spreij, Nonnegative matrix factorization an I-divergence alternating minimization, Linear Algebra Appl. 416 (2006) 270–287.
[6] D. Ho, P. van Dooren, Nonnegative matrix factorizations with fixed row and column sums, Linear Algebra Appl. (2007).
[7] D. Lee, S. Sueng, Algorithms for nonnegative matrix factorization, Adv. Neural Inf. Process. Syst. 13 (2001) 556–562.
[8] D. Lee, S. Sueng, Learning the parts of object by nonnegative matrix factorization, Nature 401 (1999) 788–791.
[9] V.P. Pauca, J. Piper, R.J. Plemmons, Nonnegative matrix factorization for spectral data analysis, Linear Algebra Appl. 416 (2006) 29–47.
[10] L.R. Rabiner, B. H Juang, An introduction to hidden Markov models, IEEE ASSP Mag. (Jun) (1986) 4–16.
[11] J. Van den Hof, System theory and system identification of compartmental systems, Ph.D. Dissertation Rijksuniversiteit Groningen, 1996.