Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

Matrix Factorization and Stochastic State Representations

Bart Vanluyten, Jan C. Willems and Bart De Moor

Abstract— **Given a two-point finite valued process, we consider the problem of finding an underlying two-point state process such that the output at a certain time instant is a probabilistic function of the state at the same time instant. This problem is related to the hidden Markov realization problem for finite valued processes. It is shown that the problem is equivalent to the algebraic problem of decomposing a square nonnegative matrix P as $VAV^\top$ with A and V nonnegative. Both multiplicative update formulas and a heuristic approach, are proposed for the solution of this decomposition problem. A simulation example shows the effectiveness of the proposed methods.**

I. INTRODUCTION

Consider a finite valued stochastic process y defined on the time axis $\mathbb{N}$ that admits a hidden Markov representation. This means that there exists an underlying Markov chain x such that the output $y(t)$ is a probabilistic function of the state $x(t)$. The *realization problem* for hidden Markov models [6] has not been completely solved yet. Even if y is known to be representable by a hidden Markov model, no algorithm has yet been devised to produce the underlying Markov chain x and the probabilistic function of the chain that produces the process y.

In this paper we consider a problem which is related to the hidden Markov realization problem. Given two random variables y^- and y^+ which both take values from the finite set $\mathbb{Y}$ with probability measure $P(y^-, y^+)$, find two random variables x^- and x^+ both with values from a finite set $\mathbb{X}$, with $|\mathbb{X}|$ as small as possible, such that

$$P(y^-, y^+|x^-, x^+) = P(y^-|x^-)P(y^+|x^+),$$

and $P(y^+|x^+) = P(y^-|x^-)$.

This problem is related to the realization problem for finite valued processes of length 2. Indeed, let $y = y(1), y(2)$ be a two-point process which is the output of a hidden Markov model, and take $y^- = y(1)$ and $y^+ = y(2)$ then the random variables x^- and x^+ can be interpreted as underlying state variables $x(1)$ and $x(2)$ respectively. The random variables x^-, x^+, y^- and y^+ are completely described by the measures $P(x^-)$, $P(x^+|x^-)$ and $P(y^-|x^-) = P(y^+|x^+)$, which correspond precisely to the state distribution at time instant 1, the probability of going from a given state to another state, and the probability of observing a certain output in a given state. This problem is only a first step of a general problem of finding a state process for general processes instead of just two-point processes.

Bart Vanluyten, Jan C. Willems and Bart De Moor are with the Electrical Engineering Department, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. Bart Vanluyten is a Research Assistant with the fund for Scientific Research-Flanders (FWO-Vlaanderen).

This problem is not to be confused with the problem where we have two random variables y^- and y^+ with probability measure $P(y^-, y^+)$, which take values from $\mathbb{Y}^-$ and $\mathbb{Y}^+$ respectively and where we want to find a random variable x with values from a finite set $\mathbb{X}$ such that y^- is conditionally independent of y^+ given x, i.e.

$$P(y^-, y^+|x) = P(y^-|x)P(y^+|x).$$

This problem, its solution based on the Nonnegative Matrix Factorization and its link with the hidden Markov realization problem is discussed in [2].

The rest of the paper is organized as follows. In Section II, we introduce the notations for hidden Markov models, in Section III, we formulate the problem of finding underlying state variables $x(1)$ and $x(2)$ for the process $y(1), y(2)$ and translate the problem into the algebraic problem of finding matrices V and A such that a given square matrix P equals $P = VAV^\top$. In Section IV, we give iterative formulas and in Section V a heuristic method for the solution of the algebraic problem. In Section VI, we perform an experiment showing the quality of the approach.

The following notation is used throughout the paper. If X is a matrix, then $X_{i:k,j:l}$ denotes the submatrix of X formed by the i-th to the k-th row and by the j-th to the l-th column of X. With X_{ij} we mean the i, j-th element of X, and with $X_{:,j}$ and $X_{i,:}$, we mean the j-th column and i-th row respectively.

II. HIDDEN MARKOV MODELS

Consider a stochastic process y defined on the time axis $\mathbb{N}$ taking values from a finite set $\mathbb{Y}$, called the output alphabet, with $|\mathbb{Y}|$ the cardinality of $\mathbb{Y}$. Denote by $\mathbb{Y}^*$ the set of all finite strings with symbols from the set $\mathbb{Y}$ (including the empty string) and by $\mathrm{y} = \mathrm{y}_1\mathrm{y}_2\ldots\mathrm{y}_{|\mathrm{y}|}$ a sequence from $\mathbb{Y}^*$, where $|\mathrm{y}|$ denotes the length of y. Let $\mathcal{P} : \mathbb{Y}^* \mapsto [0, 1]$ be string probabilities, defined as $\mathcal{P}(\mathrm{y}) := P(y(1) = \mathrm{y}_1, y(2) = \mathrm{y}_2, \ldots, y(|\mathrm{y}|) = \mathrm{y}_{|\mathrm{y}|})$. Of course, the string probabilities satisfy $\mathcal{P}(\phi) = 1$ and $\sum_{\mathrm{y}\in\mathbb{Y}} \mathcal{P}(\mathrm{y}\mathrm{y}) = \mathcal{P}(\mathrm{y})$[1].

A *Mealy hidden Markov model* (HMM) is defined as $(\mathbb{X}, \mathbb{Y}, \Pi, \pi(1))$, where

- $\mathbb{X}$ with $|\mathbb{X}| < \infty$ is the state alphabet, and $\mathbb{Y}$ is the output alphabet;
- $\pi(1)$ is a row vector in $\mathbb{R}_+^{|\mathbb{X}|}$ with $\pi(1)e = 1$, where $e := \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^\top$;
- Π is a mapping from $\mathbb{Y}$ to $\mathbb{R}_+^{|\mathbb{X}|\times|\mathbb{X}|}$ with the matrix $\Pi_{\mathbb{X}} := \sum_{\mathrm{y}\in\mathbb{Y}} \Pi(\mathrm{y})$ such that $\Pi_{\mathbb{X}} e = e$.

[1]With $\mathrm{y}\mathrm{y}$, we mean the concatenation of the string y with the symbol y. Concatenation of two strings is defined analogously.

One can think of an underlying state process x which generates the output process y. The process x takes values from the finite set $\mathbb{X}$ with cardinality $|\mathbb{X}|$. Without loss of generality, we take $\mathbb{X} = \{1, 2, \ldots, |\mathbb{X}|\}$. The element $\Pi(\mathtt{y})_{ij}$ is then equal to $P(x(t+1) = j, y(t) = \mathtt{y} | x(t) = i)$, the probability of going from state i to state j while producing the output symbol $\mathtt{y}$. The element $\pi(1)_i$ is equal to $P(x(1) = i)$, the initial distribution of the underlying state process.

In this paper, we consider the *Moore hidden Markov model*, which is a more structured case of the Mealy hidden Markov model. In a Moore HMM, the generation of the next state and the generation of the output are independent. For a Moore HMM it holds that there exists a matrix $\Pi_{\mathbb{X}}$ and a mapping β from $\mathbb{Y}$ to $\mathbb{R}_+^{|\mathbb{X}|}$ such that for each $\mathtt{y} \in \mathbb{Y}$ it holds that

$$\Pi(\mathtt{y}) = \mathrm{diag}(\beta(\mathtt{y}))\Pi_{\mathbb{X}},$$

where $\mathrm{diag}(\cdot)$ is the diagonal matrix with the elements of the vector $\cdot$ on its diagonal. The element $(\Pi_{\mathbb{X}})_{ij}$ is then equal to $P(x(t+1) = j | x(t) = i)$, the probability of going from state i to state j. The element $\beta(\mathtt{y})_i$ is equal to $P(y(t) = \mathtt{y} | x(t) = i)$, the probability of observing the symbol $\mathtt{y}$ given that the present state is equal to i. Suppose we have an ordering $(\mathtt{y}_k, k = 1, 2, \ldots, |\mathbb{Y}|)$ of the output symbols of the set $\mathbb{Y}$, then the matrix B is defined as $B = \left[\begin{array}{ccc} \beta(\mathtt{y}_1) & \ldots & \beta(\mathtt{y}_{|\mathbb{Y}|}) \end{array} \right]$. A Moore HMM is described by $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, \beta, \pi(1))$ or $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, B, \pi(1))$.

In the *(Moore) realization problem*, we are given the output string probabilities $\mathcal{P}$ and the problem is to find a Moore HMM that realizes $\mathcal{P}$, which means that for all $\mathtt{y} = \mathtt{y}_1\mathtt{y}_2 \ldots \mathtt{y}_{|\mathtt{y}|} \in \mathbb{Y}^*$, it holds that $\mathcal{P}(\mathtt{y}) = \pi(1)\,\mathrm{diag}(\beta(\mathtt{y}_1))\Pi_{\mathbb{X}}\,\mathrm{diag}(\beta(\mathtt{y}_2))\Pi_{\mathbb{X}} \ldots \mathrm{diag}(\beta(\mathtt{y}_{|\mathtt{y}|}))\Pi_{\mathbb{X}} e$.

A Moore realization $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, \beta, \pi(1))$ of $\mathcal{P}$ is called *minimal* if for any other Moore realization $(\mathbb{X}', \mathbb{Y}, \Pi'_{\mathbb{X}}, \beta', \pi'(1))$ of $\mathcal{P}$, it holds that $|\mathbb{X}| \leq |\mathbb{X}'|$.

If it holds for all $\mathtt{y} \in \mathbb{Y}^*$ that $\sum_{\mathtt{y} \in \mathbb{Y}} \mathcal{P}(\mathtt{y}\mathtt{y}) = \mathcal{P}(\mathtt{y})$ then the process is called *stationary*[2]. Because of the fact that $\sum_{\mathtt{y} \in \mathbb{Y}} \mathcal{P}(\mathtt{y}\mathtt{y}) = \mathcal{P}(\mathtt{y})$ is due to consistency, we have for stationary processes that

$$\sum_{\mathtt{y} \in \mathbb{Y}} \mathcal{P}(\mathtt{y}\mathtt{y}) = \sum_{\mathtt{y} \in \mathbb{Y}} \mathcal{P}(\mathtt{y}\mathtt{y}).$$

A hidden Markov model which realizes a stationary process, has the property that the state distribution is equal at every time instant $\pi(1) = \pi(2) = \ldots = \pi(t) = \pi$ where π equals the equilibrium state distribution, i.e.

$$\pi\Pi_{\mathbb{X}} = \pi.$$

In the next section, we consider a problem which is analogous to the minimal Moore realization problem for two-point processes (i.e. the realization problem where the data are the string probabilities of strings up to length 2): given the output probabilities of all strings of length 2, find an underlying state process of length 2, such that the output at a certain time instant is a probabilistic function of the state at the same time instant.

[2]Indeed, $\sum_{\mathtt{y} \in \mathbb{Y}} \mathcal{P}(\mathtt{y}\mathtt{y})$ is equal to $P(y(2) = \mathtt{y}_1, y(3) = \mathtt{y}_2, \ldots, y(|\mathtt{y}|+1) = \mathtt{y}_{|\mathtt{y}|})$, and by imposing this to be equal to $P(y(1) = \mathtt{y}_1, y(2) = \mathtt{y}_2, \ldots, y(|\mathtt{y}|) = \mathtt{y}_{|\mathtt{y}|})$ for all $\mathtt{y} \in \mathbb{Y}^*$, stationarity is imposed.

III. STATE REPRESENTATION OF TWO-POINT PROCESSES

In this section, we introduce a matrix P which contains probabilities of length-2-strings. In case the string probabilities come from an underlying hidden Markov model, there exist a relation between the matrix P and the system parameters of the underlying model. This relation allows us to translate the problem of finding a two-point state process for a two-point output process, into a matrix problem.

Suppose we have an ordering $(\mathtt{y}_k, k = 1, 2, \ldots, |\mathbb{Y}|)$ of the output symbols of the set $\mathbb{Y}$. Let P be the $|\mathbb{Y}| \times |\mathbb{Y}|$ matrix with k, l-th element $\mathcal{P}(\mathtt{y}_k\mathtt{y}_l)$, where $\mathtt{y}_k\mathtt{y}_l$ denotes the concatenation of the symbols $\mathtt{y}_k$ and $\mathtt{y}_l$. If the underlying process is stationary, it holds that the row sum of P is equal to its column sum, i.e. $Pe = P^\top e$.

We now derive the relation between the system matrices and the matrix P for the case where P contains the string probabilities of an underlying Moore hidden Markov model $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, B, \pi(1))$. One can easily see that

$$\begin{aligned} & P(y(1) = \mathtt{y}_k, y(2) = \mathtt{y}_l) \\ = & \sum_{i,j} P(y(1) = \mathtt{y}_k, y(2) = \mathtt{y}_l | x(1) = i, x(2) = j) P(x(1) = i, x(2) = j) \\ = & \sum_{i,j} P(y(1) = \mathtt{y}_k | x(1) = i) P(y(2) = \mathtt{y}_l | x(1) = i) P(x(1) = i, x(2) = j) \\ = & (B^\top \,\mathrm{diag}(\pi(1))\Pi_{\mathbb{X}} B)_{k,l}, \end{aligned}$$

from which we conclude that

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

The problem of finding an underlying state process for a two point output process, is now equivalent to the problem of finding for a given nonnegative $P \in \mathbb{R}^{|\mathbb{Y}| \times |\mathbb{Y}|}$ with $= e^\top P e = 1$, nonnegative matrices $B \in \mathbb{R}^{|\mathbb{X}| \times |\mathbb{Y}|}$ and $\Pi_{\mathbb{X}} \in \mathbb{R}^{|\mathbb{X}| \times |\mathbb{X}|}$ and a nonnegative vector $\pi(1) \in \mathbb{R}^{|\mathbb{X}|}$, with $|\mathbb{X}|$ as small as possible, such that

$$\begin{aligned} P &= B^\top \,\mathrm{diag}(\pi(1))\Pi_{\mathbb{X}} B, \\ Be &= e, \\ \Pi_{\mathbb{X}} e &= e, \\ \pi(1) e &= 1. \end{aligned}$$

In fact it suffices to solve the problem of finding nonnegative matrices $V \in \mathbb{R}^{|\mathbb{Y}| \times |\mathbb{X}|}$ and $A \in \mathbb{R}^{|\mathbb{X}| \times |\mathbb{X}|}$, with $|\mathbb{X}|$ as small as possible, such that

$$P = V A V^\top.$$

The matrices B and $\Pi_{\mathbb{X}}$ and the vector $\pi(1)$ can then be calculated as

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

The formulas imply that if $y = y(1), y(2)$ is stationary, i.e. $P(y(1)) = P(y(2))$ or $Pe = P^\top e$, then $x = x(1), x(2)$

is also stationary, i.e. $\pi(1) = \pi(2) = \pi$, with $\pi\Pi_{\mathbb{X}} = \pi$. Indeed, from $Pe = P^\top e$ or $B^\top A'Be = B^\top (A')^\top Be$, we find that $A'e = (A')^\top e$ if B has full row rank, i.e. if the decomposition is minimal. Now we can calculate

$$\begin{aligned}\pi(1)\Pi_{\mathbb{X}} &= (A'e)^\top (\mathrm{diag}(A'e))^{-1} A' \\ &= e^\top A' = (A'e)^\top \\ &= \pi(1).\end{aligned}$$

IV. NONNEGATIVE MATRIX FACTORIZATION

In the previous section, we reduced the problem of finding an underlying state process for two-point processes to the problem of finding a *minimal* factorization $P = VAV^\top$, where minimal means that the size of A is as small as possible. In this section, we give multiplicative update formulas to solve this problem approximately.

This problem is analogous to the standard Nonnegative Matrix Factorization problem which was introduced in [1] and which has a lot of applications in data mining. In the standard nonnegative matrix factorization problem, one is interested in decomposing a given matrix $M \in \mathbb{R}^{m_1 \times m_2}$ into a product $M = VH$. The smallest inner dimension for which such a factorization exists is called the *positive rank* (pos−rank) of the matrix M. One can show that $0 \leq \mathrm{rank}(M) \leq \mathrm{pos{-}rank}(M) \leq \min(m_1, m_2)$. There exist matrices M for which only trivial decompositions exist, $M = IM$ and $M = MI$. In [5] such matrices are called *prime*.

We call the minimal dimension of A for which an exact decomposition $P = VAV^\top$ exists, the *Markov rank* of the matrix P. It is intuitively clear that

$$\begin{aligned}0 \leq \mathrm{rank}(P) &\leq \mathrm{pos{-}rank}(P) \\ &\leq \mathrm{markov{-}rank}(P) \leq \min(m_1, m_2).\end{aligned}$$

The exact nonnegative matrix factorization problem is very hard in general. There is not even a method to determine the minimal inner dimension (i.e. the positive rank of M) for which a positive factorization exists. Therefore, Lee and Sueng proposed to chose an inner dimension and then solve the problem approximately $M \approx VH$ by an optimization based approach [4]. As cost function to quantify the quality of approximations, they take either the squared Euclidean distance

$$||A - B||^2 = \sum_{ij} (A_{ij} - B_{ij})^2$$

or the Kullback-Leibler divergence (which is a measure of the extent to which B approximates A)

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

For both cost functions, they derive multiplicative update rules which monotonically improve the quality of the approximation. In [3], these update rules are interpreted as alternating minimization procedures and stability properties are investigated.

In the rest of this section, we will analogously derive multiplicative update formulas for the factorization problem $P \approx VAV^\top$. As in [4], the dimension of the matrix A is chosen by the user. We derive only formulas with the Kullback-Leibler divergence as cost function, as this distance is most appropriate for the approximation of probability measures. However, it is also possible to derive multiplicative update formulas for the factorization problem in the Euclidian distance. We will publish these formulas elsewhere. Checking whether there exists an interpretation of the update rules as an alternating minimization procedure, analogous to [3], belongs to our future research plans.

Now we consider the problem

Problem 1: Minimize $g(A, V) = D(P||VAV^\top)$ for a given size of A with respect to A and V, subject to the constraints $A, V \geq 0$.

The derivatives of the cost function with respect to the elements $V_{k,i}$ and $A_{i,j}$ of the matrices A and V can be calculated as

$$\begin{aligned}\frac{\partial g}{\partial A_{ij}}(A, V) &= -\sum_\mu \sum_\nu V_{\mu i} V_{\nu j} \frac{P_{\mu\nu}}{(VAV^\top)_{\mu\nu}} \\ &\quad + \sum_\mu \sum_\nu V_{\mu i} V_{\nu j}, \\ \frac{\partial g}{\partial V_{ki}}(A, V) &= \sum_\lambda \sum_\nu V_{\nu\lambda} A_{i\lambda} + V_{\nu\lambda} A_{\lambda i} \\ &\quad - \sum_\lambda \sum_\nu V_{\nu\lambda} A_{i\lambda} \frac{P_{k\nu}}{(VAV^\top)_{k\nu}} + V_{\nu\lambda} A_{\lambda i} \frac{P_{\nu k}}{(VAV^\top)_{\nu k}}.\end{aligned}$$

A simple rule for updating A and V which reduces the Kullback-Leibler distance can be written as

$$\begin{aligned}A_{ij} &\leftarrow A_{ij} - \mu_{ij} \frac{\partial g}{\partial A_{ij}}(A, V) \\ V_{ki} &\leftarrow V_{ki} - \nu_{ki} \frac{\partial g}{\partial V_{ki}}(A, V)\end{aligned}$$

If all μ_{ij} and ν_{ki} are equal to the same small positive number, this is equivalent to conventional gradient descent. As long as this number is sufficiently small, the update reduces $D(P||VAV^\top)$. The problem with the gradient descent method is that the choice of the step size is difficult. If the step size is too small, we have slow convergence. On the other hand, if the step size is too large, the matrices can become negative or it is possible that the squared distance does not decrease.

Analogous to the method proposed in [4], we propose to take the steps sizes equal to

$$\begin{aligned}\mu_{ij} &= \frac{A_{ij}}{\sum_\mu \sum_\nu V_{\mu i} V_{\nu j}}, \\ \nu_{ki} &= \frac{V_{ki}}{\sum_\lambda \sum_\nu V_{\nu\lambda} A_{i\lambda} + V_{\nu\lambda} A_{\lambda i}},\end{aligned}$$

which gives

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

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

By this choice of step size which is dependent of the optimization variables, we do not have a gradient descent algorithm anymore, but instead we have update formulas where each step consists of multiplication with a factor. In particular, it is straightforward to notice that this factor is equal to unity if $P = VAV^\top$, so perfect reconstruction is a fixed point of the algorithm. Notice also that this factor is always nonnegative, so if one takes nonnegative initial values for A and V, then the updated values will always be nonnegative. In addition, it can be proven that these iterative update formulas converge to a local minimum of the cost function. The proof will be presented elsewhere.

It is interesting to further investigate the factorization problem in the Kullback-Leibler divergence with Markov rank equal to 1. In that case, we look for a vector v and a scalar a such that $P \approx vav^\top$. This is equivalent with solving $\min_{a,v} \quad D(P||vav^\top)$ or

$$\min_{a,v} \quad -\sum_{\mu\nu} P_{\mu\nu} \log([vav^\top]_{\mu\nu}) + \sum_{\mu\nu} [vav^\top]_{\mu\nu}.$$

The solution can be found by setting the derivatives equal to 0:

$$\begin{aligned} \frac{\partial D(P||vav^\top)}{\partial v_k} &= -\sum_\nu \frac{P_{k\nu} + P_{\nu k}}{v_k} + 2\sum_\nu a v_\nu = 0 \\ \frac{\partial D(P||vav^\top)}{\partial a} &= -\sum_{\mu\nu} \frac{P_{\mu\nu}}{a} + \sum_{\mu\nu} v_\mu v_\nu = 0. \end{aligned}$$

Because $\sum_{\mu\nu} P_{\mu\nu} = 1$, we find that

$$\begin{aligned} a &= (\sum_\nu v_\nu)^2, \\ v_k &= \frac{\sum_\nu (P_{k\nu} + P_{\nu k})}{2a \sum_\nu v_\nu}. \end{aligned}$$

We can always normalize the solution such that $a = 1$. We then find

$$v_k = \frac{\sum_\nu (P_{k\nu} + P_{\nu k})}{2}.$$

This result is interesting. It says that a matrix P can be approximated by $P \approx vv^\top$ in an optimal way (w.r.t the Kullback-Leibler divergence) by taking v equal to the mean of the row and column sum of P, i.e. $v = \frac{1}{2}(Pe + P^\top e)$. This means that the probability of observing $\mathtt{y}_k$ at time instant 1 and $\mathtt{y}_l$ at time instant 2 is approximated as the product of the mean probability of observing $\mathtt{y}_k$ times the mean probability of observing $\mathtt{y}_l$, i.e.

$$\mathcal{P}(\mathtt{y}_k \mathtt{y}_l) \approx \frac{P(y(1) = \mathtt{y}_k) + P(y(2) = \mathtt{y}_k)}{2} \frac{P(y(1) = \mathtt{y}_l) + P(y(2) = \mathtt{y}_l)}{2}.$$

In case $y = y(1), y(2)$ is stationary, then $P(y(1) = \mathtt{y}_k, y(2) = \mathtt{y}_l)$ is approximated as $P(y(1) = \mathtt{y}_k) P(y(2) = \mathtt{y}_l)$, which is equal to the product of the marginal distributions.

V. A HEURISTIC APPROACH TO NONNEGATIVE MATRIX FACTORIZATION

In this section we propose a heuristic approach for the calculation of the decomposition $P = B^\top \operatorname{diag}(\pi(1)) \Pi_{\mathbb{X}} B$ without making use of the iterative update formulas. In this approach we start with a full Markov rank decomposition i.e. a decomposition where the size of $\Pi_{\mathbb{X}}$ is maximal and then merge rows and columns of $\Pi_{\mathbb{X}}$ and B in an appropriate way until the decomposition has the required inner dimension.

The method starts with the trivial full Markov rank decomposition of the matrix P i.e.

$$P = (B^{|\mathbb{Y}|})^\top \operatorname{diag}(\pi(1)^{|\mathbb{Y}|}) \Pi_{\mathbb{X}}^{|\mathbb{Y}|} B^{|\mathbb{Y}|},$$

where

$$\begin{aligned} B^{|\mathbb{Y}|} &= I, \\ \pi(1)^{|\mathbb{Y}|} &= Pe, \\ \Pi_{\mathbb{X}}^{|\mathbb{Y}|} &= (\operatorname{diag}(Pe))^{-1} P, \end{aligned}$$

where I denotes the unit matrix.

We now propose a heuristic procedure to find an approximate decomposition of Markov rank $n-1$, given an (approximate) decomposition of Markov rank equal to n (at the beginning of the algorithm, the approximation is exact and $n = |\mathbb{Y}|$),

- Suppose $\pi(1)_i^n$ is the smallest and $\pi(1)_j^n$ is the second smallest element of the vector $\pi(1)^n$. The vector $\pi(1)^{n-1}$ is then calculated by replacing the i-th element of $\pi(1)^n$ by $\pi(1)_i^n + \pi(1)_j^n$ and omitting the j-th element.
- The matrix $\Pi_{\mathbb{X}}^{n-1}$ can be calculated from the matrix $\Pi_{\mathbb{X}}^n$ by
  1) replacing the i-th column of $\Pi_{\mathbb{X}}^n$ by the sum of the i-th and the j-th column, and omitting the j-th column. Call the resulting matrix M.
  2) replacing the i-th row of M by $\frac{\pi(1)_i^n}{\pi(1)_i^n + \pi(1)_j^n} M_{i,:} + \frac{\pi(1)_j^n}{\pi(1)_i^n + \pi(1)_j^n} M_{j,:}$, and omitting the j-th row of M.
- The matrix B^{n-1} can be found by replacing the i-th row of B^n by $\frac{\pi(1)_i^n}{\pi(1)_i^n + \pi(1)_j^n} B_{i,:}^n + \frac{\pi(1)_j^n}{\pi(1)_i^n + \pi(1)_j^n} B_{j,:}^n$, and omitting the j-th row of B^n.

One can choose to stop the algorithm when a pre-decribed order is reached or when the smallest element of the vector $\pi(1)^n$ is larger than a certain threshold.

We now give some intuition into this heuristic approach. One step of the heuristic algorithm goes from a hidden Markov model with n states to a hidden Markov model with $n-1$ states by merging the two states i and j with the smallest initial probabilities. The output probabilities of the new state are equal to the weighted mean output probabilities of the states i and j where the weighting factors

are the relative initial probabilities of the merged states. The probability to go from an arbitrary state h to the merged state is equal to the probability to go from state h to state i plus the probability to go from state h to state j. The probability to go from the merged state to another state g is equal to the weighted sum of the probability to go from state i to state g and the probability to go from state j to state g, where the weighting factors are the relative initial probabilities of the merged states. Finally, the probability to go from the merged state to itself is equal to the weighted sum of the probability to go from state i to state i or state j, and the probability to go from state j to state i or state j, where the weighting factors are again equal to the relative initial probabilities of the merged states.

If the process is stationary, the initial state distribution is equal to the equilibrium distribution. In that case the heuristic method is expected to perform best, as it merges states with the smallest equilibrium distribution instead of the smallest initial distribution.

We now prove that, if we start with a probability matrix P and continue the proposed heuristic approach until there is only one state anymore, we find that P is approximated by the vector containing the row sum of P (i.e. the measure $P(y(1))$) multiplied with the transpose of the same vector. This means that we find $P = (B^1)^\top \operatorname{diag}(\pi^1_{t-1})\Pi^1_{\mathbb{X}} B^1$, with

$$\begin{aligned} B^1 &= (Pe)^\top, \\ \pi^1_{t-1} &= 1, \\ \Pi^1_{\mathbb{X}} &= 1. \end{aligned}$$

We prove this fact for the case where $P \in \mathbb{R}^{3\times 3}$. The general proof is analogous. In the 3×3 case, we start form

$$P = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \pi_1 & & \\ & \pi_2 & \\ & & \pi_3 \end{bmatrix} \Pi^3_{\mathbb{X}} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix},$$

where we used π_i as a shorthand notation for $\pi(1)_i$. We suppose that $\pi_1 \geq \pi_2 \geq \pi_3$, such that we find after one step

$$P \approx \begin{bmatrix} 1 & & \\ & \frac{\pi_1}{\pi_2+\pi_3} & \\ & \frac{\pi_2}{\pi_2+\pi_3} & \end{bmatrix} \begin{bmatrix} \pi_1 & & \\ & \pi_2+\pi_3 & \end{bmatrix} \Pi^2_{\mathbb{X}} \begin{bmatrix} 1 & & \\ & \frac{\pi_2}{\pi_2+\pi_3} & \frac{\pi_3}{\pi_2+\pi_3} \end{bmatrix},$$

and after two steps, we find

$$P \approx \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} [\pi_1+\pi_2+\pi_3]\, \Pi^1_{\mathbb{X}} \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix},$$

where one can easily see that $\Pi^1_{\mathbb{X}} = 1$, which proves the statement for $P \in \mathbb{R}^{3\times 3}$.

If the underlying hidden Markov model is stationary, then the row sums of the matrix P are equal to the column sums, i.e. $Pe = P^\top e$, which means that the heuristic algorithm applied on the probability matrix P converges to the product of the marginal distributions.

VI. SIMULATION EXAMPLE

In this simulation example we consider an output process with $\mathbb{Y} = \{a, b, \ldots, j\}$, and suppose that the string probabilities of all strings of length 2 are given and the problem is to find an underlying two-point state process and a probabilistic mapping from the state process to the output process.

We suppose there is an ordering $(\mathtt{Y}_1 = a, \mathtt{Y}_2 = b, \ldots, \mathtt{Y}_{10} = j)$ on the output set $\mathbb{Y}$. Now the string probabilities can be stacked in the matrix P as described in Section III, i.e. $P_{kl} = \mathcal{P}(\mathtt{Y}_k\mathtt{Y}_l)$.

In this 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 $P = B^\top \operatorname{diag}(\pi)\Pi_{\mathbb{X}} B$ where $B = \begin{bmatrix} \beta(a) & \beta(b) & \beta(c) & \ldots & \beta(j) \end{bmatrix}$, $\Pi_{\mathbb{X}}$ and π come from a stationary Moore HMM $(\mathbb{X}, \mathbb{Y}, \Pi_{\mathbb{X}}, \beta, \pi)$ with $\mathbb{X} = \{1, 2, \ldots, 5\}$ and

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

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

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

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

We use the iterative update algorithm of Section IV to compute optimal approximations with respect to the Kullback-Leibler divergence with Markov rank equal to $1, 2, \ldots, 10$. As initial values for the iterative algorithm we use random nonnegative matrices. As stopping rule, we use the Kullback-Leibler divergence between the approximation at iteration step i and the approximation at step $i+1$. The algorithm stops if this distance is smaller than 10^{-8}. In Table I we show the number of steps until convergence for the different Markov ranks. Finally, we compute approximations with the heuristic approach of Section V, also of Markov rank equal to $1, 2, \ldots, 10$.

On Figure 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 Markov rank. At the same figure we plot the Kullback-Leibler divergence between the original matrix and the heuristic approximation also for all possible Markov ranks.

TABLE I

NUMBER OF ITERATIONS FOR THE MULTIPLICATIVE UPDATE METHOD MINIMIZING THE KULLBACK-LEIBLER DIVERGENCE.

	Order 1	Order 2	Order 3	Order 4	Order 5
Number of iterations	1687	1055	3004	2694	1804
	Order 6	Order 7	Order 8	Order 9	Order 10
Number of iterations	1817	962	151	191	3

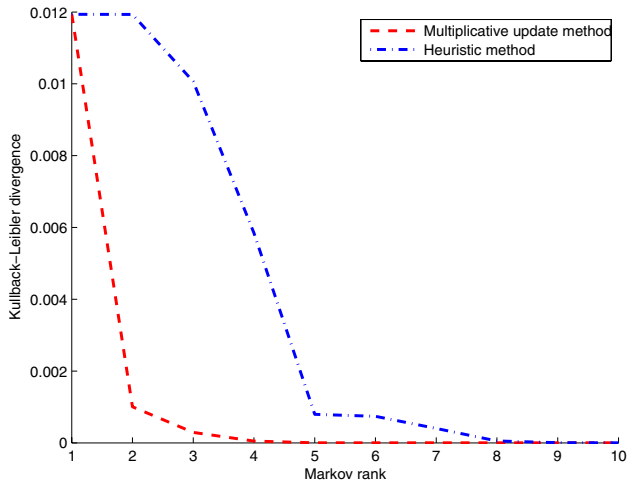

Fig. 1. Kullback-Leibler divergence between the true matrix P and its optimal (w.r.t the Kullback-Leibler divergence) approximation of Markov rank $1, 2, \ldots, 10$ computed with the iterative algorithm of Section IV (-.-.) and between the true matrix P and its approximation of Markov rank $1, 2, \ldots, 10$ computed with the heuristic algorithm of Section V (- - -).

Notice that the iterative method performs much better than the heuristic approach. The distance is almost equal to 0 for Markov ranks 5 to 10. This makes sense as the matrix P was generated using an underlying hidden Markov model of order 5. Notice also that the Kullback-Leibler divergence between P and its optimal rank 1 approximation (w.r.t the Kullback-Leibler divergence) is equal to the Kullback-Leibler divergence between P and the heuristic approximation of rank 1. This is no coincidence, as we have have proven that the rank 1 approximation of P found with the Kullback-Leibler minimization method is equal to the product of the marginals of P in case the underlying hidden Markov model is stationary and the same holds for the heuristic approach.

To show further the quality of the approximations, we give in Table II the true output probabilities of a selection (due to space limitations) of length-2 strings and compare them with the probabilities found with the Kullback-Leibler minimalisation method of order $5, 4, \ldots 1$. We conclude that the approximation of the matrix P as $VAV^\top$ works well which allows to conclude that the modeling of two-point string probabilities with a hidden Markov model works well.

TABLE II

STRING PROBABILITIES FOR STRINGS OF LENGTH 2.

Sequence	Exact	Order 5	Order 4	Order 3	Order 2	Order 1
aa	0.0396	0.0396	0.0397	0.0430	0.0430	0.0362
ab	0.0193	0.0193	0.0190	0.0191	0.0191	0.0207
ac	0.0149	0.0149	0.0150	0.0152	0.0152	0.0156
ad	0.0116	0.0116	0.0116	0.0114	0.0114	0.0137
ae	0.0113	0.0113	0.0114	0.0110	0.0110	0.0128
af	0.0094	0.0094	0.0095	0.0094	0.0094	0.0114
ag	0.0098	0.0099	0.0100	0.0098	0.0098	0.0118
ah	0.0161	0.0161	0.0158	0.0153	0.0153	0.0184
ai	0.0128	0.0127	0.0128	0.0122	0.0122	0.0139
aj	0.0454	0.0454	0.0453	0.0437	0.0437	0.0357

VII. CONCLUSION

In this paper we considered the problem of finding an underlying two-point Markov state process for a given two-point finite valued process, such that the output at a certain time instant is a probabilistic function of the state at that time instant. It was shown that the problem is equivalent to the algebraic problem of decomposing a nonnegative matrix P as the product $VAV^\top$ with A nonnegative and of minimal dimension and V nonnegative. Both multiplicative update formulas and a heuristic approach, were proposed for the solution of this decomposition problem. This article is only a first step in the problem of finding a realization of a hidden Markov model from given string probabilities. The extension of the proposed methods to a time axis $(1, 2, \ldots, T)$ or $\mathbb{N}$ is part of our future research.

ACKNOWLEDGMENTS

The SISTA research program is supported by:

Research Council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/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.0553.06 (VitamineD), research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants,GBOU (McKnow), Eureka-Flite2; Belgian Federal Science Policy Office: IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling', 2002-2006) ; PODO-II (CP/40: TMS and Sustainability); EU: FP5-Quprodis; ERNSI; Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, Mastercard

REFERENCES

[1] D. Lee and S. Sueng, Learning the Parts of Object by Non-Negative Matrix Factorization, *Nature*, vol. 401, 1999, pp 788-791.

[2] L. Finesso and P. Spreij, Approximate realization of hidden Markov chains, *Proceedings of the 2002 IEEE Information Theory Workshop*, Bangalore, India, 2002.

[3] L. Finesso and P. Spreij, Approximate nonnegative matrix factorization via alternating minimization, *Proceedings of the 16th International Symposium on MTNS*, Leuven, 2004, pp 223-252.

[4] D. Lee and S. Sueng, Algorithms for Non-Negative Matrix Factorization, *Advances in Neural Information Processing Systems*, vol. 13, 2001, pp 556-562.

[5] J. van den Hof, System theory and system identification of compartmental systems, PhD Thesis *Rijksuniversiteit Groningen*, 1996.

[6] B.D.O. Anderson, The realization problem for hidden Markov models, *Mathematics of Control, Signals, and Systems*, vol. 12-1, 1999, pp 80-120.