**Pattern Analysis and Applications manuscript No.**
(will be inserted by the editor)

# Dimensionality Reduction and Topographic Mapping of Binary Tensors

**Jakub Mažgut · Peter Tiňo · Mikael Bodén · Hong Yan**

Received: date / Accepted: date

**Abstract** In this paper, a decomposition method for binary tensors, generalized multi-linear model for principal component analysis (GMLPCA) is proposed. To the best of our knowledge at present there is no other principled systematic framework for decomposition or topographic mapping of binary tensors. In the model formulation, we constrain the natural parameters of the Bernoulli distributions for each tensor element to lie in a sub-space spanned by a reduced set of basis (principal) tensors. We evaluate and compare the proposed GMLPCA technique with existing real-valued tensor decomposition methods in two scenarios: (1) in a series of controlled experiments involving synthetic data; (2) on a real world biological dataset of DNA sub-sequences from different functional regions, with sequences represented by binary tensors. The experiments suggest that the GMLPCA model is better suited for modeling binary tensors than its real-valued counterparts. Furthermore, we extended our GMLPCA model to the semi-supervised setting by forcing the model to search for a natural parameter subspace that represents a user specified compromise between the modelling quality and the degree of class separation.

**Keywords** Tensor decomposition · Tucker model · Binary data · Topographic mapping

J. Mažgut
Faculty of Informatics and Information Technologies
Slovak University of Technology
81219 Bratislava, Slovakia
E-mail: mazgut@gmail.com

P. Tiňo
School of Computer Science
University of Birmingham
Birmingham B15 2TT, United Kingdom
E-mail: P.Tino@cs.bham.ac.uk

M. Bodén
School of Chemistry and Molecular Biosciences and
School of Information Technology and Electrical Engineering
The University of Queensland, QLD 4072, Australia
E-mail: m.boden@uq.edu.au

H. Yan
The Dept. Electronic Engineering
City University of Hong Kong
Kowloon, Hong Kong
E-mail: h.yan@cityu.edu.hk

## 1 Introduction

At present an increasing number of data processing tasks involve manipulation of multi-dimensional objects, known also as tensors, where the elements are to be addressed by more than two indices. In many practical problems such as gait [1], hand postures [2] or face recognition [3], hyperspectral image processing [4] and text documents analysis [5], the data tensors are specified in a high-dimensional space. Applying pattern recognition or machine learning methods directly to such data spaces can result in high computational and memory requirements, as well as poor generalization. To address this "curse of dimensionality" a wide range of decomposition methods have been introduced to compress the data while capturing the 'dominant' trends. Making the learning machines operate on this compressed data space may not only boost their generalization performance but crucially can also increase their interpretability.

Decomposition techniques such as principal component analysis (PCA) [6] were designed to decompose data objects in the form of vectors. For tensor decomposition, the data items need to be first vectorized before the analysis can be applied. Besides higher computational and memory requirements, the vectorization of

data tensors breaks the higher order dependencies presented in the natural data structure that can potentially lead to more compact and useful representations [1]. New methods capable of processing multi-dimensional tensors in their natural structure have been introduced for real-valued tensors [1,5,7], nonnegative tensors [8,9] and symmetric tensors [10]. Such techniques, however, are not suitable for processing binary tensors. Yet, binary tensors arise in many real world applications such as gait recognition [1], document analysis [5] or graph objects represented by adjacency tensors. In this paper we introduce and verify model based methods for unsupervised and semi-supervised binary tensor decomposition that explicitly take into an account the binary nature of such data. In particular:

1. We extend the model of [11] for decomposition of binary vectors into a methodology for decomposition of binary tensors of any (finite) order. We show that even though the original model is non-linear in parameters, the strong *linear* algebraic structure of the Tucker model for tensor decomposition can be superimposed on the parameter space of our model, allowing us to preserve the efficient linear nature of parameter updates introduced in [11].
2. We extend our model for unsupervised decomposition of binary tensors to the semi-supervised setting where the knowledge of tensor labels can be utilized to the degree specified by the user.

The paper is organized as follows: Section 2 introduces notation and basic tensor algebra. Section 3 briefly discusses the problem of reduced rank representation of real-valued tensors. A model based formulation for binary data decomposition with iterative update scheme to maximize the model's log-likelihood is presented in sections 4 and 5. Section 6 contains experiments on synthetic and biological datasets. Finally, section 7 discusses an extension of GMLPCA to semi-supervised learning and section 8 summarizes the results and concludes the work.

## 2 Notation and Basic Tensor Algebra

An $N$-th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ can be thought of as an $N$-dimensional array of real numbers in programming languages. It is addressed by $N$ indices $i_n$ ranging from 1 to $I_n$, $n = 1, 2, ..., N$. A rank-1 tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ can be obtained as an outer product of $N$ non-zero vectors $\mathbf{u}^{(n)} \in \mathbb{R}^{I_n}$, $n = 1, 2, ..., N$: $\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$. In other words, for a particular index setting $(i_1, i_2, ..., i_N) \in \Upsilon = \{1, 2, ..., I_1\} \times \{1, 2, ..., I_2\} \times ... \times \{1, 2, ..., I_N\}$, we have

$$\mathcal{A}_{i_1, i_2, ..., i_N} = \prod_{n=1}^{N} u_{i_n}^{(n)}, \tag{1}$$

where $u_{i_n}^{(n)}$ is the $i_n$-th component of the vector $\mathbf{u}^{(n)}$. Slightly abusing mathematical notation, we will often write the index $N$-tuples $(i_1, i_2, ..., i_N) \in \Upsilon$ using vector notation $\mathbf{i} = (i_1, i_2, ..., i_N)$, so that instead of writing $\mathcal{A}_{i_1, i_2, ..., i_N}$ we write $\mathcal{A}_{\mathbf{i}}$.

A tensor can be multiplied by a matrix (2nd order tensor) using $n$-mode products: The $n$-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ by a matrix $U \in \mathbb{R}^{J \times I_n}$ is a tensor $(\mathcal{A} \times_n U)$ with entries $(\mathcal{A} \times_n U)_{i_1, ..., i_{n-1}, j, i_{n+1}, ..., i_N} = \sum_{i_n=1}^{I_n} \mathcal{A}_{i_1, ..., i_{n-1}, i_n, i_{n+1}, ..., i_N} \cdot U_{j, i_n}$, for some $j \in \{1, 2, ..., J\}$.

Consider now an orthonormal basis $\{\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, ..., \mathbf{u}_{I_n}^{(n)}\}$ for the $n$-mode space $\mathbb{R}^{I_n}$. The (column) vectors $\mathbf{u}_k^{(n)}$ can be stored as columns of the basis matrix $U^{(n)} = (\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, ..., \mathbf{u}_{I_n}^{(n)})$. Any tensor $\mathcal{A}$ can be decomposed into the product $\mathcal{A} = \mathcal{Q} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 ... \times_N U^{(N)}$, with expansion coefficients stored in the $N$th order tensor $\mathcal{Q} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$. The expansion of $\mathcal{A}$ can also be written as

$$\mathcal{A} = \sum_{\mathbf{i} \in \Upsilon} \mathcal{Q}_{\mathbf{i}} \cdot (\mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ ... \circ \mathbf{u}_{i_N}^{(N)}). \tag{2}$$

In other words, tensor $\mathcal{A}$ is expressed as a linear combination of $\prod_{n=1}^{N} I_n$ rank-1 basis tensors $(\mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ ... \circ \mathbf{u}_{i_N}^{(N)})$. In addition, from orthonormality of the basis sets, the tensor $\mathcal{Q}$ of expansion coefficients can be obtained as $\mathcal{Q} = \mathcal{A} \times_1 (U^{(1)})^T \times_2 (U^{(2)})^T \times_3 ... \times_N (U^{(N)})^T$.

This is also known as the Tucker model for decomposing real tensors: any tensor $\mathcal{A}$ (of the given size) is expressed as a linear combination of rank-1 (basis) tensors obtained as outer products of the corresponding basis vectors. Besides the Tucker model, another widely used tensor decomposition technique is related to the more restricted PARAFAC model [12]. PARAFAC model can be viewed as a special case of the Tucker model, where the tensor $\mathcal{Q}$ of expansion coefficients is diagonal, every mode has an equal number $I$ of basis vectors yielding $I$ rank-1 basis tensors (the $i$-th rank-1 basis tensor is equal to $(\mathbf{u}_i^{(1)} \circ \mathbf{u}_i^{(2)} \circ ... \circ \mathbf{u}_i^{(N)})$). For our GMLPCA model we use the more flexible Tucker model considering the results of Wang and Ahuja [13]. The experiments with real-valued tensors showed that Tucker model achieved lower reconstruction errors than the PARAFAC model with the same compression ratio.

## 3 Reduced Rank Representations of Tensors

Several approaches have been proposed for reduced rank representations of tensors (e.g. [14–16]). For example, one can assume that a smaller number of basis tensors in the expansion (2) are sufficient to approximate all tensors in a given dataset:

$$\mathcal{A} \approx \sum_{\mathbf{i} \in K} \mathcal{Q}_{\mathbf{i}} \cdot (\mathbf{u}^{(1)}_{i_1} \circ \mathbf{u}^{(2)}_{i_2} \circ ... \circ \mathbf{u}^{(N)}_{i_N}), \tag{3}$$

where $K \subset \mathcal{Y}$. In other words, tensors in a given dataset can be found 'close' to the $|K|$-dimensional hyperplane in the tensor space spanned by the basis tensors $(\mathbf{u}^{(1)}_{i_1} \circ \mathbf{u}^{(2)}_{i_2} \circ ... \circ \mathbf{u}^{(N)}_{i_N})$, $\mathbf{i} \in K$. Then the tensor $\mathcal{A}$ can be represented through expansion coefficients $\mathcal{Q}_{\mathbf{i}}$, $\mathbf{i} \in K$.

Note that the orthonormality of basis $\{\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{I_n}\}$ for the $n$-mode space $\mathbb{R}^{I_n}$ can be relaxed. It can be easily shown that as long as for each mode $n = 1, 2, ..., N$, the vectors $\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{I_n}$ are linearly independent, the basis tensors $(\mathbf{u}^{(1)}_{i_1} \circ \mathbf{u}^{(2)}_{i_2} \circ ... \circ \mathbf{u}^{(N)}_{i_N})$, $\mathbf{i} \in \mathcal{Y}$ will be linearly independent as well. If the $n$-mode space basis are orthonormal, the tensor decomposition is known as the Higher-Order Singular Value Decomposition (HOSVD) [14]. It has to be said that extending matrix (2nd-order tensor) decompositions such as SVD to higher-order tensors is not an easy matter. Familiar concepts such as rank become ambiguous and more complex. However, the main purpose of the decomposition remains unchanged: rewrite a tensor as a sum of rank-1 tensors.

One of the early attempts to extend the traditional PCA method for multidimensional tensors is represented by the GPCA model[17]. The GPCA model is an extension of PCA for 2*nd*-order tensors and similarly to PCA aims to maximize the captured variation in the (projected) data. Further generalizations of PCA and GPCA for arbitrary order real-valued tensors were proposed in the same year (2008): Xu et al. introduced the concurrent subspace analysis (CSA)[18] and Lu et al. proposed the multilinear PCA (MPCA)[1]. Both models are based on the Tucker concept and use an iterative estimation scheme to fit the model parameters[1]. Besides the basic MPCA model, many other extensions were introduced, uncorrelated multilinear PCA (UMPCA)[19], robust multilinear PCA[20] and a version for non-negative tensors[9]. Further details about multilinear subspace learning models can be found in [21], where Tucker based models are denoted as *tensor-to-tensor projections* and PARAFAC based models as *tensor-to-vector projections*.

[1] The updating formulas of CSA and MPCA are similar, the only difference being that MPCA subtracts the mean from the data tensors.

While much work has been done in the context of PCA-style decompositions of real-valued tensors, no formalism exists as yet for decomposing binary tensors. Binary tensors occur naturally in many applications where the value $\mathcal{A}_{i_1,i_2,...,i_N}$ indicates presence or absence of the feature related to the index $(i_1, i_2, ..., i_N)$. For example, in graph theory, a 2nd-order tensor $\mathcal{A}$ (called adjacency matrix) codes a graph by imposing $\mathcal{A}_{i_1,i_2} = 1$ if and only if there is an arc from node $i_1$ to node $i_2$, $\mathcal{A}_{i_1,i_2} = 0$ otherwise. We present a framework that is a generalization of the binary probabilistic principal component analysis to tensor data.

## 4 The Model

Our GMLPCA model for binary tensor decomposition and topographic mapping is based on an extension of the multi-linear Tucker model (2) for real-valued tensorial data. The extension is analogous to the generalization of linear models for exponential family of distributions. We use the Tucker model as a multi-linear 'predictor' and logistic function as a link function to link the real-valued multi-linear 'predictions' with response variables, in our case binary elements of data tensors. A probabilistic framework is used to formally define the model and to derive rules for the parameter estimation.

Consider an $N$th-order tensor $\mathcal{A} \in \{0, 1\}^{I_1 \times I_2 \times ... \times I_N}$. Assume we are given a set of $M$ such binary tensors $\mathcal{D} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_M\}$. Each element $\mathcal{A}_{m,\mathbf{i}}$ of the tensor $\mathcal{A}_m$, $m = 1, 2, ..., M$, is assumed to be (independently) Bernoulli distributed with parameter (mean) $p_{m,\mathbf{i}}$:

$$P(\mathcal{A}_{m,\mathbf{i}} | p_{m,\mathbf{i}}) = p_{m,\mathbf{i}}^{\mathcal{A}_{m,\mathbf{i}}} \cdot (1 - p_{m,\mathbf{i}})^{1 - \mathcal{A}_{m,\mathbf{i}}}. \tag{4}$$

The Bernoulli distribution can be equivalently parametrized through log-odds (natural parameter) $\theta_{m,\mathbf{i}} \in \mathbb{R}$, so that the canonical link function linking the natural parameter with the mean is the logistic function

$$p_{m,\mathbf{i}} = \sigma(\theta_{m,\mathbf{i}}) = \frac{1}{1 + e^{-\theta_{m,\mathbf{i}}}}. \tag{5}$$

For each data tensor $\mathcal{A}_m$, $m = 1, 2, ..., M$, we have

$$P(\mathcal{A}_m | \theta_m) = \prod_{\mathbf{i} \in \mathcal{Y}} \sigma(\theta_{m,\mathbf{i}})^{\mathcal{A}_{m,\mathbf{i}}} \cdot \sigma(-\theta_{m,\mathbf{i}})^{1 - \mathcal{A}_{m,\mathbf{i}}}. \tag{6}$$

We collect all the parameters $\theta_{m,\mathbf{i}}$ in a tensor $\Theta \in \mathbb{R}^{M \times I_1 \times I_2 \times ... \times I_N}$ of order $N + 1$. Assuming the data tensors in $\mathcal{D}$ are independently generated, the model

log-likelihood reads

$$\mathcal{L}(\Theta) = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\Upsilon} \mathcal{A}_{m,\mathbf{i}} \log\sigma(\theta_{m,\mathbf{i}}) + (1-\mathcal{A}_{m,\mathbf{i}}) \log\sigma(-\theta_{m,\mathbf{i}}). \quad (7)$$

So far the values in the parameter tensor $\Theta$ were unconstrained. To discover a low dimensional structure in the data, we employ the multi-liner Tucker model to constrain all the $N$-th order parameter tensors $\theta_m \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ (one for each data tensor $\mathcal{A}_m$) to lie in the subspace spanned by the reduced set of basis tensors $(\mathbf{u}^{(1)}_{r_1}\circ\mathbf{u}^{(2)}_{r_2}\circ\ldots\circ\mathbf{u}^{(N)}_{r_N})$, where $r_n \in \{1,2,...,R_n\}$, and $R_n \leq I_n$, $i = 1,2...,N$. The indices $\mathbf{r} = (r_1, r_2, ..., r_N)$ take values from the set $\rho = \{1,...,R_1\}\times\{1,...,R_2\}\times\ldots\times\{1,...,R_N\}$. Note that there is no explicit pressure in the model to ensure (for each mode) independence of the basis vectors. However, in practice, the optimized model parameters always represented independent basis vectors, as dependent basis vectors would lead to dependent basis tensors, implying smaller than intended rank of the tensor decomposition.

We further allow for an $N$-th order bias tensor $\Delta \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, so that the parameter tensors $\theta_m$ are constrained onto an affine space. Using (2) we get

$$\theta_m = \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}} \cdot (\mathbf{u}^{(1)}_{r_1}\circ\mathbf{u}^{(2)}_{r_2}\circ\ldots\circ\mathbf{u}^{(N)}_{r_N}) + \Delta \quad (8)$$

so that by

$$\theta_{m,\mathbf{i}} = \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}} \cdot (\mathbf{u}^{(1)}_{r_1}\circ\mathbf{u}^{(2)}_{r_2}\circ\ldots\circ\mathbf{u}^{(N)}_{r_N})_{\mathbf{i}} + \Delta_{\mathbf{i}} = \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}} \cdot \prod_{n=1}^{N} u^{(n)}_{r_n,i_n} + \Delta_{\mathbf{i}}. \quad (9)$$

## 5 Algorithm

In this section, we present only the final update rules for fitting the model parameters with a minimum of extra formalism. A full derivation of the updating rules can be found in appendix.

We use the trick of [11] and take advantage of the fact that while the model log-likelihood (7) is not convex in the parameters, it is convex in any parameter, if the others are kept fixed. This leads to an iterative scheme where the model parameters are fitted alternating between least square updates for basis vectors $\mathbf{u}^{(n)}_{r_n}$, expansion coefficients $Q_{m,\mathbf{r}}$ and bias tensor $\Delta$. While one set of the parameters is updated, the others are held fixed. This procedure is repeated until the log-likelihood converges to a desired degree of precision. The updates lead to monotonic increases in the log-likelihood.

### 5.1 Updates for $n$-mode space basis

Holding the bias tensor $\Delta$ and the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, $m = 1,2,...,M$, $\mathbf{r}\in\rho$ fixed, we obtain a update rule for the $n$-mode space basis $\{\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{R_n}\}$. For each $n$-mode and its coordinate $j \in \{1,2,\ldots,I_n\}$, the basis vectors are updated by solving linear system:

$$\sum_{t=1}^{R_n} u^{(n)}_{t,j}\ \mathcal{K}^{(n)}_{q,t,j} = \mathcal{S}^{(n)}_{q,j}, \quad (10)$$

where

$$\mathcal{S}^{(n)}_{q,j} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\Upsilon_{-n}} (2\mathcal{A}_{m,[\mathbf{i},j|n]} - 1 - \Psi_{m,[\mathbf{i},j|n]}\Delta_{[\mathbf{i},j|n]})\mathcal{B}^{(n)}_{m,\mathbf{i},q}, \quad (11)$$

$$\mathcal{K}^{(n)}_{q,t,j} = \sum_{m=1}^{M} \sum_{\mathbf{r}\in\rho_{-n}} \mathcal{Q}_{m,[\mathbf{r},t|n]} \times \sum_{\mathbf{i}\in\Upsilon_{-n}} \Psi_{m,[\mathbf{i},j|n]}\ \mathcal{B}^{(n)}_{m,\mathbf{i},q} \prod_{s=1,s\neq n}^{N} u^{(s)}_{r_s,i_s}, \quad (12)$$

$$\mathcal{B}^{(n)}_{m,\mathbf{i},q} = \sum_{\mathbf{r}\in\rho_{-n}} \mathcal{Q}_{m,[\mathbf{r},q|n]} \cdot \prod_{s=1,s\neq n}^{N} u^{(s)}_{r_s,i_s}, \quad (13)$$

$\Psi$ denotes $(\tanh\frac{\theta_{m,\mathbf{i}}}{2})/\theta_{m,\mathbf{i}}$, and $q = 1,2,\ldots,R_n$. Note that the updates for coefficients of basis vectors for different mode $n$ and its coordinates $j \in \{1,2,\ldots,I_n\}$ are conveniently decoupled.

### 5.2 Updates for expansion coefficients

When updating the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, the bias tensor $\Delta$ and the basis sets $\{\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{R_n}\}$ for all $n$ modes $n = 1,2,...,N$ are kept fixed to their current values. The update rule for expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$ of the $m$-th input tensor $\mathcal{A}_m$ can be obtained by solving a set of linear equations

$$\mathcal{T}_{\mathbf{v},m} = \sum_{\mathbf{r}\in\rho} \mathcal{P}_{\mathbf{v},\mathbf{r},m}\ \mathcal{Q}_{m,\mathbf{r}}, \quad (14)$$

where

$$\mathcal{T}_{\mathbf{v},m} = \sum_{\mathbf{i}\in\Upsilon} (2\mathcal{A}_{m,\mathbf{i}} - 1 - \Psi_{m,\mathbf{i}}\ \Delta_{\mathbf{i}})\ C_{\mathbf{v},\mathbf{i}}, \quad (15)$$

$$\mathcal{P}_{\mathbf{v},\mathbf{r},m} = \sum_{\mathbf{i}\in\Upsilon} \Psi_{m,\mathbf{i}}\ C_{\mathbf{v},\mathbf{i}}\ C_{\mathbf{r},\mathbf{i}}, \quad (16)$$

$C_{\mathbf{r},\mathbf{i}}$ denotes $\prod_{n=1}^{N} u^{(n)}_{r_n,i_n}$ and $\mathbf{v}\in\rho$ is a basis index. In terms of these equations, the expansion coefficients updates for different input tensors are conveniently decoupled.

### 5.3 Updates for the bias tensor

As before, holding the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, $m = 1, 2, ..., M$, $\mathbf{r} \in \rho$, and the basis sets $\{\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, ..., \mathbf{u}_{R_n}^{(n)}\}$ for all $n$ modes $n = 1, 2, ..., N$ fixed, we obtain a simple update rule for the bias tensor:

$$\Delta_{\mathbf{j}} = \frac{\sum_{m=1}^{M} 2\mathcal{A}_{m,\mathbf{j}} - 1 - \Psi_{m,\mathbf{j}} \cdot \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}}\ C_{\mathbf{r},\mathbf{j}}}{\sum_{m=1}^{M} \Psi_{m,\mathbf{j}}}. \quad (17)$$

### 5.4 Decomposing Unseen Binary Tensors

Note that our model is not generative, however, it is straightforward to find expansion coefficients for an $N$-th order tensor $\mathcal{A}' \in \{0,1\}^{I_1 \times I_2 \times ... \times I_N}$ not included in the training set $\mathcal{D}$. One simply needs to solve for expansion coefficients in the natural parameter space, given that the parameters are confined onto the affine subspace of the tensor parameter space found in the training phase. Recall that the affine subspace is determined by the bias tensor $\Delta$ and the basis sets $\{\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, ..., \mathbf{u}_{R_n}^{(n)}\}$, one for each $n$ mode, $n = 1, 2, ..., N$. These are kept fixed.

The log-likelihood (7) to be maximized with respect to the expansion coefficients stored in tensor $\mathcal{Q}$ reads

$$\mathcal{L}(\mathcal{Q}; \mathcal{A}') = \sum_{\mathbf{i},\ s.t.\ \mathcal{A}'_{\mathbf{i}}=1} \log\sigma\left(\sum_{\mathbf{r}\in\rho} \mathcal{Q}_{\mathbf{r}}\ C_{\mathbf{r},\mathbf{i}} + \Delta_{\mathbf{i}}\right)$$
$$+ \sum_{\mathbf{i},\ s.t.\ \mathcal{A}'_{\mathbf{i}}=0} \log\sigma\left(-\sum_{\mathbf{r}\in\rho} \mathcal{Q}_{\mathbf{r}}\ C_{\mathbf{r},\mathbf{i}} - \Delta_{\mathbf{i}}\right). \quad (18)$$

Any optimization technique can be used. The quantities $C_{\mathbf{r},\mathbf{i}}$ and $\Delta_{\mathbf{i}}$ are constants given by the trained model. The tensor $\mathcal{Q}$ can be initialized by first finding the closest data tensor from the training data set $\mathcal{D}$ to $\mathcal{A}'$ in the Hamming distance sense,

$$m(\mathcal{A}') = \underset{m=1,2,...,M}{\arg\min} \sum_{\mathbf{i}\in\mathcal{Y}} |\mathcal{A}'_{\mathbf{i}} - \mathcal{A}_{m,\mathbf{i}}|,$$

and then setting the initial value of $\mathcal{Q}$ to the expansion coefficient tensor of $\mathcal{A}_{m(\mathcal{A}')}$.

When using gradient ascent,

$$\mathcal{Q}_{\mathbf{v}} \leftarrow \mathcal{Q}_{\mathbf{v}} + \eta\ \frac{\partial\ \mathcal{L}(\mathcal{Q}; \mathcal{A}')}{\partial\ \mathcal{Q}_{\mathbf{v}}},$$

the updates take the form

$$\mathcal{Q}_{\mathbf{v}} \leftarrow \mathcal{Q}_{\mathbf{v}} + \eta \sum_{\mathbf{i}\in\mathcal{Y}} C_{\mathbf{v},\mathbf{i}} \left[\mathcal{A}'_{\mathbf{i}} - \sigma\left(\sum_{\mathbf{r}\in\rho} \mathcal{Q}_{\mathbf{r}}\ C_{\mathbf{r},\mathbf{i}} + \Delta_{\mathbf{i}}\right)\right], \quad (19)$$

where $\eta > 0$.

## 6 Experiments

In this section, we compare our proposed generalized multilinear model for principal component analysis of binary tensors (GMLPCA) with one matrix decomposition method logistic principal component analysis (LPCA) and three existing real-valued tensor decomposition methods, namely tensor latent semantic indexing model (TensorLSI)[5], multilinear principal component analysis model (MPCA)[1] and uncorrelated multilinear principal component analysis (UMPCA)[19]. We incorporate the LPCA into the experiment to point out the advantage of tensor decomposition methods over the classical vector methods. Note that an application of LPCA to binary tensors requires their reshaping into vectors.

The comparison is performed in two different scenarios. First, we evaluate the ability of the models to compress (and reconstruct) synthetic binary data tensors in a controlled set of experiments. Second, we illustrate our method on a real set of biological DNA subsequences (represented as binary tensors) originating from different functional regions of genomic sequences.

### 6.1 Synthetic Data

In order to evaluate the ability of the model to find a compact data representation and reconstruct the compressed data, we generated several datasets of binary tensors from underlying subspaces of the natural parameter space. Below, we describe generation of the synthetic binary tensors, give an overall outline of the experiments and summarize the results.

#### *6.1.1 Generating process*

Our goal is to generate a set of $M$ binary tensors $\mathcal{D} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_M\}$, where each element $\mathcal{A}_{m,\mathbf{i}}$ of $\mathcal{A}_m \in \{0,1\}^{I_1, I_2, ..., I_N}$ is independently Bernoulli distributed with natural parameter $\theta_{m,\mathbf{i}}$ and the natural parameter tensor $\theta_m$ lies in a sub-space spanned by a set of linearly independent basis tensors $\{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_R\}$, $\mathcal{B}_r \in \{-1,1\}^{I_1 \times I_2 \times ... \times I_N}$, $r = 1, 2, ..., R$. Given such basis, each synthetic binary tensor $\mathcal{A}_m$ is generated as follows: First, a tensor $\theta_m$ containing Bernoulli natural parameters is constructed as a (random) linear combination of bases, $\theta_m = \sum_{r=1}^{R} \alpha_{mr}\mathcal{B}_r$, where the elements $\alpha_{m_r} \in \mathbb{R}$ of the mixing vector $\boldsymbol{\alpha}_m = (\alpha_{m_1}, \alpha_{m_2}, ..., \alpha_{m_R})$, are sampled from a uniform distribution over a given support. The elements $\mathcal{A}_{m,\mathbf{i}}$ of the binary data tensor $\mathcal{A}_m$ are then sampled from the Bernoulli distribution parametrized by $\theta_{m,\mathbf{i}}$ (see (4) and (5)):

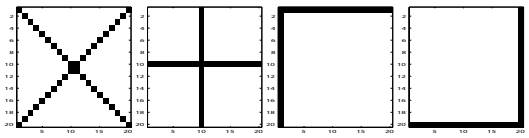


**Fig. 1** An example of basis tensors spanning a 4-dimensional Bernoulli natural parameter space.

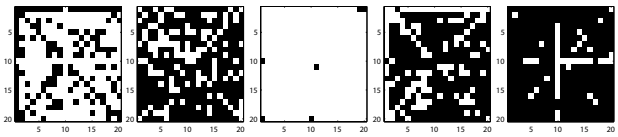


**Fig. 2** A sample of randomly generated binary tensors from the Bernoulli natural parameter space spanned by the bases shown in figure 1.

$$\mathcal{A}_{m,\mathbf{i}} \sim P(\mathcal{A}_{m,\mathbf{i}}|\theta_{m,\mathbf{i}}) = \sigma(\theta_{m,\mathbf{i}})^{\mathcal{A}_{m,\mathbf{i}}} \cdot \sigma(-\theta_{m,\mathbf{i}})^{1-\mathcal{A}_{m,\mathbf{i}}}. \tag{20}$$

To illustrate that it is non-trivial to discern the underlying natural parameter subspace from the sample binary tensors, we randomly sampled and visualized 5 data tensors from Bernoulli natural parameter space spanned by bases shown in figure 1. The binary tensors are shown in figure 2.

### *6.1.2 Outline of the Experiments*

In the next section we will use tensor decomposition to analyze a large-scale set of biological sequences represented through sparse second order binary tensors ($N = 2$) of sizes around $(I_1, I_2) = (30, 250)$. In this section we verify our method in a set of controlled experiments employing synthetically generated 2nd-order binary tensors of size $(I_1, I_2) = (30, 30)$. For the experiment we generated 5 data sets, each containing $M = 3,000$ binary tensors, from a Bernoulli natural parameter spaces spanned by 40 linearly independent basis tensors. Each data set was sampled from a different natural parameter subspace. From each data set we hold out one-third $(1,000)$ binary tensors as a test set and let the models find the latent subspace on the remaining $(2,000)$ tensors (training set).

After training the models, tensors that were not included in training (hold-out set) were "compressed" by projecting them onto the principal subspace and thus their low dimensional representations in the natural parameter space were obtained. To evaluate the amount of preserved information, the compressed representations would need to be reconstructed back into the original binary tensor space. Note that since the models we consider represent binary tensors through continuous values in the natural parameter space (GMLPCA, LPCA), or in $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ (other models), a straightforward deterministic reconstruction in the binary space is not appropriate. We use the area under the ROC curve (AUC) designed to compare different real-valued predictions of binary data. One way of determining the AUC value is to calculate the normalized Wilcoxon-Mann-Whitney statistic which is equal to AUC [22]. If we identify $\{x_1, x_2 \dots x_J\}$ as the model prediction outputs for all nonzero elements of tensors from the test set, and $\{y_1, y_2 \dots y_K\}$ as outputs for all zero elements, the AUC value for that particular prediction (reconstruction) of the test set of tensors is equal to

$$\text{AUC} = \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} C(x_j, y_k)}{J \cdot K},$$

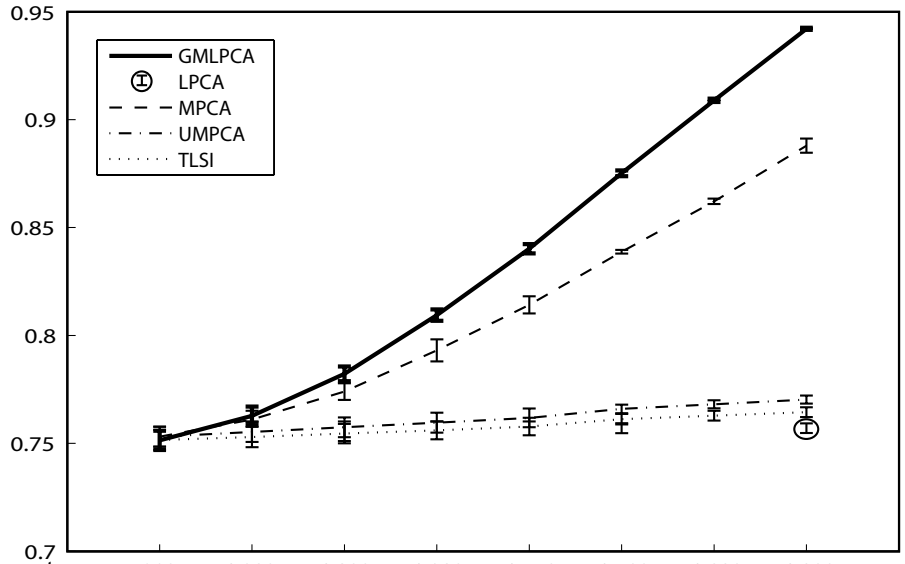
where $J$ and $K$ are the total number of nonzero and zero tensor elements in the test set, respectively, and $C$ is a scoring function

$$C(x_j, y_k) = \begin{cases} 1 & \text{if} x_j > y_k \\ 0 & \text{otherwise.} \end{cases}$$

For a fair comparison, the decomposition methods are compared based on the amount of preserved information with respect to the number of free parameters. In general, the free parameters correspond to the basis vectors and the offset. For the TensorLSI, MPCA and UMPCA the offset represents the mean tensor (used to center the data); for GMLPCA and LPCA it represents the bias tensor and vector, respectively. For GMLPCA and MPCA, the number of free parameters is equal to $\sum_{n=1}^{N} R_n \cdot I_n + \prod_{n=1}^{N} I_n$. TensorLSI with $R$ basis tensors has $R \cdot \sum_{n=1}^{N} \min\{R, I_n\} + \prod_{n=1}^{N} I_n$ free parameters. UMPCA is a PARAFAC based model and can extract up to $R \leq \min\{\min I_n, M\}$ uncorrelated features (basis tensors). The number of free parameters for UMPCA is equal to $R \cdot \sum_{n=1}^{N} I_n + \prod_{n=1}^{N} I_n$. For the LPCA the number of free parameters is equal to $(R+1) \cdot \prod_{n=1}^{N} I_n$, where $R$ is the number of basis vectors.

### *6.1.3 Results*

We summarize performance of the examined models to compress and subsequently reconstruct the sets of synthetic tensors by calculating the mean and standard deviation of AUC values across the 5 test sets of binary tensors. Reconstruction results for binary tensors generated from Bernoulli natural parameter spaces spanned by 40 bases tensors are summarized in figure 3. For the two smallest subspace sizes (strong compression), the models achieved comparable results while for the other larger subspace sizes GMLPCA clearly significantly outperformed all the other models.




| Number of free parameters | 960 | 1 080 | 1 200 | 1 320 | 1 440 | 1 560 | 1 680 | 1 800 |
|---|---|---|---|---|---|---|---|---|
| GMLPCA (row x column vectors) | 1 x 1 | 3 x 3 | 5 x 5 | 7 x 7 | 9 x 9 | 11 x 11 | 13 x 13 | 15 x 15 |
| MPCA (row x column vectors) | 1 x 1 | 3 x 3 | 5 x 5 | 7 x 7 | 9 x 9 | 11 x 11 | 13 x 13 | 15 x 15 |
| UMPCA (uncorrelated features) | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
| TensorLSI (basis tensors) | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
| LPCA (basis vectors) | - | - | - | - | - | - | - | 1 |

**Fig. 3** AUC analysis of hold-out binary tensor reconstructions obtained by the models using different number of free parameters among 5 different sets of binary tensors. Each set was generated from different Bernoulli natural parameter space spanned by 40 linearly independent basis tensors. Table under the plot describes model settings for particular number of free parameters.

### 6.2 Topographic Mapping of DNA Sequences represented as binary tensors

Bioinformatics involves the development of computational tools for systematic analysis and visualization of DNA, RNA and protein sequence data. To uncover specific regulatory circuits controlling gene expression, biologists need to first confidently map broader functional regions implanted in genomic sequence, such as *promoters*. Promoters are regions upstream of a gene, used by the transcriptional machinery. The process of expressing a gene is carefully regulated by the timely binding of both general and gene-specific regulatory proteins and RNA (or complexes thereof) to its promoter. We thus expect that promoters contain regulatory binding sites (typically 5-15 nucleotide degenerate sequence patterns). This section investigates the ability of GMLPCA to recognize in an unsupervised manner promoter sequences (forming our "positive" class).

A gene contains *exons* and *introns*, where exons primarily code for protein-forming amino acids. Not directly coding for amino acids, introns may contain important control signals for splicing the gene product. Intronic sequences may also contain sites to which a different category of regulatory proteins (and RNA) bind to modulate the efficiency of transcription. Many current state-of-the-art systems for analyzing and predicting promoter regions (e.g. [23,24]) are based on the underlying principle that sequences from different functional regions differ in a local term[2] composition. Following [23,24], we use intronic sequences as a negative set. For further information on DNA sub-sequence analysis methods, among others, we refer the reader to [25] and [26].

Based on this principle we use a suffix tree based extraction of statistically significant terms, preserving the within-class (functional regions) frequencies. Given such terms, the local term composition of a DNA sequence is obtained in the form of a binary second-order tensor (matrix), where rows represent terms, columns positions within the sequence and the binary tensor elements indicate the presence/absence of a term in the sequence at a given position.

To reveal the 'dominant trends' in a real world large-scale dataset of annotated DNA sequences, we compress the binary tensors representing the sequences into their low dimensional representations and visualize their distributions. Note that based on the underlying assumption about the differences in local term composition, we expect some separation between sequences from differ-

[2] As a term, we denote a short and widespread sequence of nucleotides that has or may have a biological significance.

ent functional regions of DNA, even though the decomposition/compression models themselves are fitted in a completely unsupervised manner.

### *6.2.1 Biological sequence data and its representation*

We use the dataset of promoter and intron (non-promoter) sequences employed in [23]. From the Database of Transcription Start Sites (DBTSS), version 5.2 [27], which includes 30,964 human promoter sequences, we extracted from each sequence a subsequence from 200bp upstream to 50bp downstream relative to the position of a transcription start site (TSS). Regulatory components primarily bind to the DNA relatively close to the TSS. The same number of intron sequences with length of 250bp were randomly selected from the Exon-Intron Database [28], release Sept.2005. To represent the sequences, we identify terms over the alphabet of nucleotides $\mathcal{N} \in \{a, c, g, t\}$ that are statistically significant longest words preserving the within-class frequencies. For this purpose we use a suffix tree construction. Typically, such a construction is guided by two main characteristics: **(1)** a 'significance criterion' used to decide whether to continue with expanding a particular suffix and **(2)** construction parameters guiding the suffix extension process. As the significance criterion we employ the Kullback-Leibler divergence between the promoter and intron class distributions, given by the candidate term $w$ and its possible extension $ws$, $s \in \mathcal{N}$, weighted by the prior distribution of the extended term $ws$. The suffix tree is built in a bottom-up fashion, starting with four leaf nodes labeled with the symbols from $\mathcal{N}$. A term $w$ is extended with a symbol $s$ if

$$P(ws) \sum_{c=\{0,1\}} P(c|ws) \log_2 \frac{P(c|ws)}{P(c|w)} \geq \epsilon_{KL},$$

where the classes of sequences from promoter regions and intron regions are denoted by $c = 1$ and $c = 0$, respectively. Size of the suffix tree depends on values of the construction parameters $\epsilon_{KL}, \epsilon_{grow} > 0$. The parameter $\epsilon_{grow}$ represents the minimal frequency of a word in the training sequences to be considered a candidate for the tree construction. More details on general principles behind suffix tree construction can be found e.g. in [29,30].

Besides the composition of specific terms, their position within the sequences may be an important factor (especially for promoter sequences aligned with respect to the TSS site). To capture both the term composition and position, we represent the DNA sequences as binary second-order tensors $\mathcal{A}$ where rows $i_1$ represent terms, columns $i_2$ positions within the sequence, and the binary tensor element $\mathcal{A}_{i_1,i_2}$ is an indicator whether the sequence represented by $\mathcal{A}$ has a term $i_1$ at position $i_2$.

An example of a real promoter sequence representation by a binary tensor is shown in figure 4. The sequence was randomly selected from the dataset.

We used the following values of the construction parameters: $\epsilon_{grow} = 5 \times 10^{-3}$ and $\epsilon_{KL} = 7 \times 10^{-5}$. This setting yields a set of 31 terms. Larger term sets (obtained using lower values of construction parameters) did not improve the separation of sequences from different function regions in the final visualizations and smaller term sets (resulting from more stringent parameter settings) made the separation weaker. Each DNA sequence was represented by a binary matrix with 31 rows and 250 columns. We compressed the binary matrices via GMLPCA using 10 principal tensors obtained as outer products of 5 column and 2 row basis vectors. The setting of 5 column and 2 row basis vectors corresponds to the smallest principal subspace for GMLPCA that lead to a significant separation of promoters from introns. To see if the other decomposition methods are capable of revealing the same level of separation with this number of principal tensors, we decomposed the data using (1) MPCA with 5 column and 2 row basis vectors; (2) UMPCA with 10 uncorrelated features; (3) TensorLSI using 10 principal tensors.

### *6.2.2 Visualizations*

All the decomposition methods represent the sequential data as 10-dimensional vectors of expansion coefficients. To visualize the distribution of such representations, we used principal component analysis (PCA) and projected the real-valued 10-dimensional expansion vectors onto the two-dimensional space defined by the two leading principal vectors. Visualizations of promoter and intron sequences decomposed via GMLPCA, TensorLSI, MPCA and UMPCA are shown in figure 5. Promoter sequences are illustrated in the plots by blue and intron sequences by red dots. The separation between promoters and introns is markedly better under the GMLPCA model than under the real-valued MPCA, UMPCA and TensorLSI methods.

For a more involved analysis, we project the 10-dimensional expansion coefficients of the GMLPCA onto the leading two-dimensional principal subspace (see figure 6). Detailed analysis of the individual sequences (not reported here) reveals that the rightmost region, populated primarily by promoter sequences, have frequent occurrences of terms GGCG, GCG, CGCG and CCGC. This indicates a high concentration of di-nucleotides CG around the TSS. These so-called CpG islands

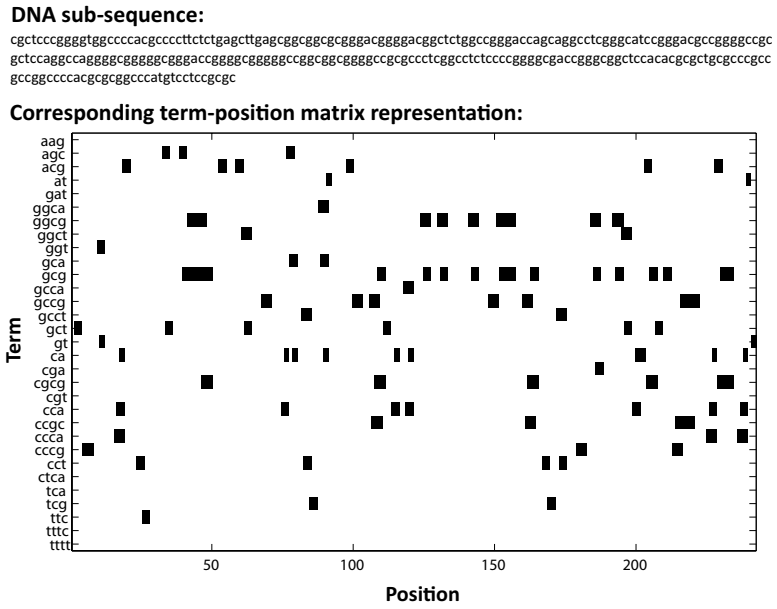



**Fig. 4** An example of representing a promoter DNA sub-sequence from real biological dataset by a binary second order tensor where rows represent terms and columns positions.

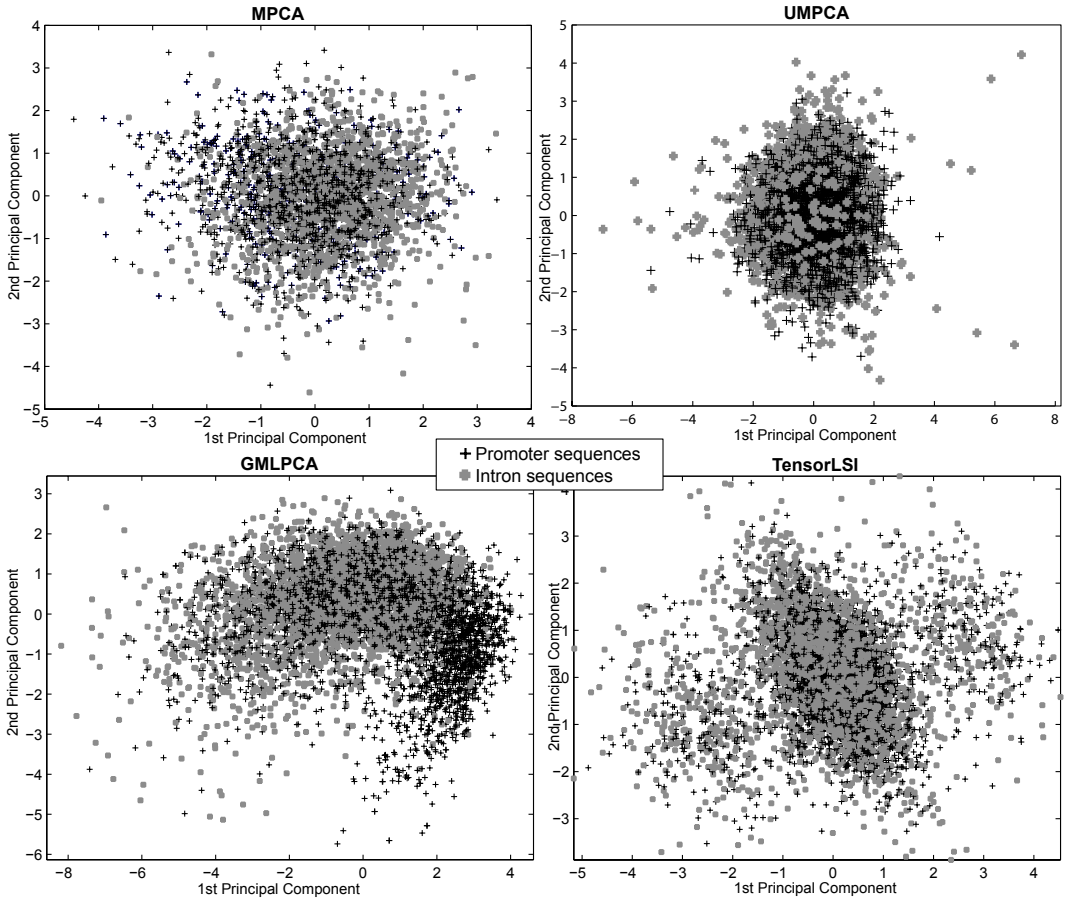



**Fig. 5** Two-dimensional PCA projections of 10% randomly sampled promoter and intron sequences from the tensor space spanned by 10 basis tensors obtained by the MPCA (upper-left plot), UMPCA (upper-right plot), GMLPCA (lower-left plot) and TensorLSI (lower-right plot) models.

are known to be associated with functional promoter regions–approximately 60% of mammalian genes [31]. More specific signals are found directly at the TSS (the band at "200"), which is known to be a key site for the RNA polymerase transcriptional machinery. GT and AG are known signals for splicing and are thus expected to occur in introns. Indeed, these words tend to occur predominately in the intron-rich regions in figure 6.

For a deeper analysis of the composition difference between promoter and intron sequences, a user interaction can be integrated into the visualization to select and visualize the term composition of interesting individual sequences. An illustrative visualization of manually selected sequences is shown in figure 6 where matrices for promoters and introns are denoted by letters P and I, respectively. Based on our previous analysis, we highlighted important terms that have a strong influence on the sequence position in the visualization space (marked with black dots). For illustration purposes, we selected two pairs of 'close' promoter sequences: the pair (P-3,P-4) is more separated from the introns than the pair (P-1,P-2). Based on the term compositions, promoter sequences P-3 and P-4 have higher occurrences of terms GGCG and GCG (regarded as a strong signal of promoter sequence by the model). A general topographic organization of the visualization plot is clearly visible, with 'close' sequences representations on the plot having 'similar' term composition structure (e.g three intron sequences I-1, I-2 and I-3, and related promoter structure in P-5).

### *6.2.3 Functional enrichment analysis of promoter sequences*

The DNA-binding sites of transcription factors are often characterized as relatively short (5-15 nucleotides) and degenerate sequence patterns. They may occur multiple times in promoters of the genes the expression of which they modulate. To further validate that GMLPCA indeed picks up biologically meaningful patterns, we searched the compressed feature space of promoters for biologically relevant structure (including that left by transcription factors). Genes that are transcribed by the same factors are often functionally similar [32]. Carrying specific biologically relevant features, suitable representations of promoters should correlate with the roles assigned to their genes. If the projection to a compressed space highlights such features, it is testament to a method's utility for processing biological sequences.

The Gene Ontology (GO; [33]) provides a controlled vocabulary for the annotation of genes, broadly categorized into terms for cellular component, biological process and molecular function. In an attempt to assign biologically meaningful labels to promoters, all sequences were mapped to gene identifiers. In cases of multiple promoters for the same identifier, we picked one sequence randomly. In cases of multiple gene identifiers for the same promoter sequence, we picked the identifier with the greatest number of annotations. Using the Gene Ontology (June 2009), we could thus assign zero or more GO terms to each promoter sequence. In total there are 8051 unique GO terms annotating 14619 promoters.

Recall that in this experiment GMLPCA decomposes binary tensors into a (linear) combination of 10 basis tensors in the Bernoulli natural parameter space. Each promoter sequence from the dataset can thus be represented by a 10-dimensional expansion coefficient vector. For visualization purposes, standard PCA is then used to project the expansion vectors into a 2-dimensional space (selected to have the highest principal values). To evaluate whether promoters deemed similar by GMLPCA are also functionally similar, we need first to design a methodology for calculating the 'distance' between each pair of promoters. Naively, one may be tempted to use the usual Euclidean distance in the 10-dimensional coordinate space of natural parameters. However, this is not correct, since **(1)** the basis tensors are not orthogonal; **(2)** they span a space of Bernoulli natural parameters that have a nonlinear relationship with the data values. To determinate the model-based 'distance' between two promoter sequences $m$ and $l$ in a principled manner, we propose to calculate a sum of average symmetrized Kullback-Leibler divergences between noise distributions for all corresponding tensor elements $\mathbf{i} \in \mathcal{Y}$:

$$D(m,l) = \sum_{\mathbf{i}\in\mathcal{Y}} \left( \frac{\mathrm{KL}[p_{m,\mathbf{i}} \,||\, p_{l,\mathbf{i}}] + \mathrm{KL}[p_{l,\mathbf{i}} \,||\, p_{m,\mathbf{i}}]}{2} \right), \tag{21}$$

where KL divergence between two Bernoulli distributions defined by their means (see (4)) is equal to

$$\mathrm{KL}[p_{m,\mathbf{i}} \,||\, p_{l,\mathbf{i}}] = \sum_{x\in\{0,1\}} P(x|p_{m,\mathbf{i}}) \log \frac{P(x|p_{m,\mathbf{i}})}{P(x|p_{l,\mathbf{i}})}. \tag{22}$$

The following test suite aims to quantify if the compressed promoter representations are biologically meaningful. In each test, we select one promoter as a reference. The test is repeated until all promoters have been selected. Given a reference promoter $m$, we label the group of all promoters $l$ within a pre-specified distance $D(m,l) < D_0$ as "positives" and all others as "negatives". Hence, the positive set of the reference

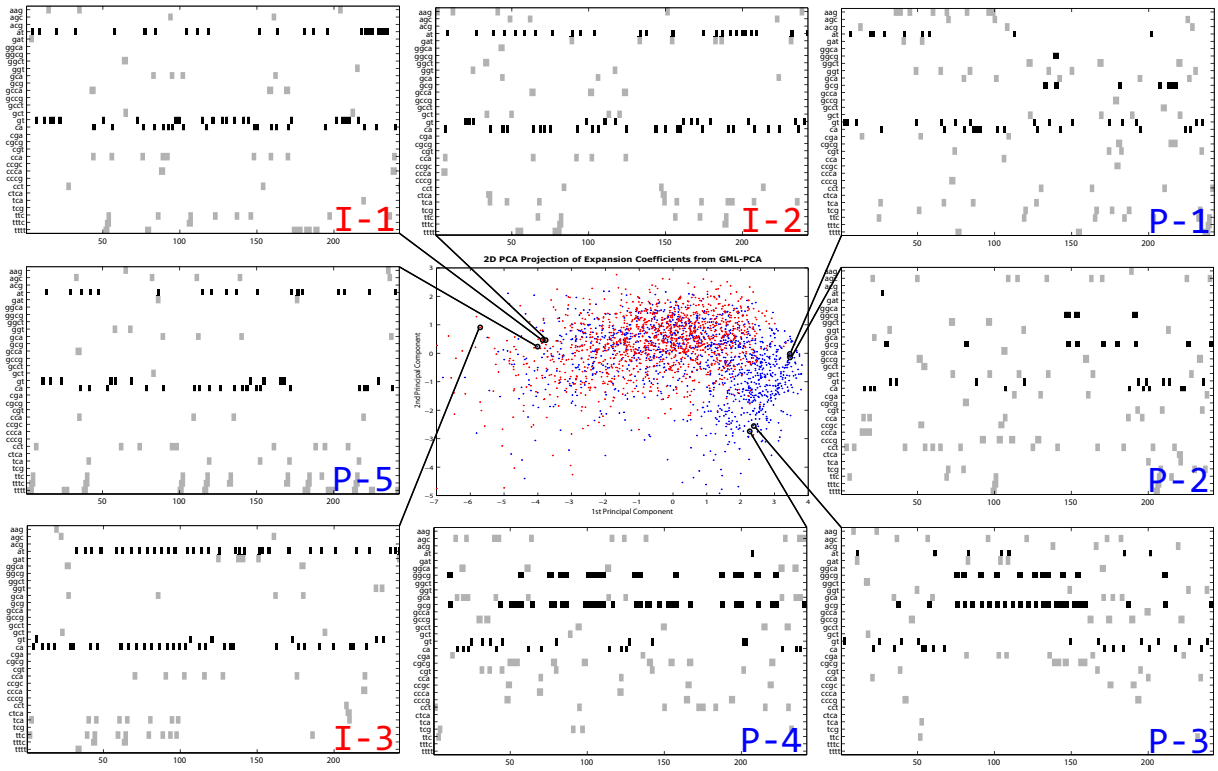



**Fig. 6** Detailed visualization of second-order binary tensors for manually selected promoter and intron sequences. Important terms that have a strong influence on the sequence coordinates in the central 2D plot are marked with black dots.

promoter $m$ reads: $S_m = \{l \mid D(m,l) < D_0\}$. In the tests we consistently use a distance of $D_0 = 25$, usually rendering over one hundred "positives". For each GO term (ultimately, in the full Gene Ontology, but in practice, we look only at those assigned to the reference promoter), Fisher's exact test resolves if it occurs more often amongst "positives" than would be expected by chance. (The null hypothesis is that the GO term is not attributed more often than by chance to the "positives".) A small $p$-value indicates that the term is "enriched" at the position of the reference promoter $m$. We adjust for multiple hypothesis testing and set the threshold at which to report a term as significant accordingly ($p < 5 \cdot 10^{-7}$). To understand the tendency of false discovery, we also repeated the tests (with the same significance threshold) after shuffling the points assigned to promoters. Re-assuringly, in no case did this permutation test identify a single GO term as significant.

In total, at the aforementioned level of significance, we found 75 GO terms that were enriched around one or more reference promoters. The observation that a subset of promoter sequences are functionally organized after decomposition into 10 basis tensors adds support to the methods' ability to detect variation at an information-rich level. More specifically, we find a number of terms that are specifically concerned with chromatin structure (that packages the DNA), e.g. GO:0000786 "Nucleosome", GO:0006333 "Chromatin assembly or disassembly" and GO:0065004 "Protein-DNA complex assembly". Interestingly, we found several enriched terms related to development, e.g. GO:0022414 "Reproductive process" and GO:0007565 "Female pregnancy". Anecdotally, we note that CpG islands (that are clearly distinct in the promoter sequence data) are associated with open DNA, leading to constitutive gene expression. Speculatively, genes associated with CpG-rich promoters need to control local chromatin. Moreover, it was recently observed that under-methylation of such otherwise methylation-prone regions is established by developmental cues [34], suggesting a link between CpG islands and development.

We use the following method to visualize the grouping of promoters assigned the same GO term $t$. Given a promoter $m$, we consider two events: $e_t$ - a promoter assigned the GO term $t$; $e_{\neq t}$ - the complement of $e_t$. The probability $P_t(e_t|S_m)$ is determined by calculating the proportion of promoters with the GO term $t$ in $S_m$. We do this by first normalizing the counts so that *a priori* (across all promoters) the probability of drawing a positive (a promoter assigned the GO term $t$) and drawing a negative (a promoter not assigned that GO term) is

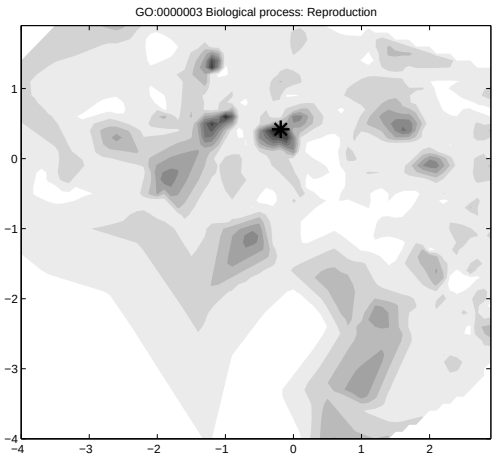



**Fig. 7** Promoter regions assigned to GO:0000003 biological process: Reproduction.

equal. Denote by $M$, $M_t$, $M_{t,S_m}$, and $M_{\neq t,S_m}$ the number of promoters, number of promoters assigned to GO term $t$, number of promoters from $S_m$ assigned to GO term $t$, and number of promoters from $S_m$ not assigned to GO term $t$, respectively. Then the counts $M_{t,S_m}$ and $M_{\neq t,S_m}$ are normalized as $\tilde{M}_{t,S_m} = (M_{t,S_m} \cdot M)/(2M_t)$ and $\tilde{M}_{\neq t,S_m} = [M_{\neq t,S_m} \cdot M]/[2(M - M_t)]$, yielding

$$P_t(e_t|S_m) = \frac{\tilde{M}_{t,S_m}}{\tilde{M}_{t,S_m} + \tilde{M}_{\neq t,S_m}}.$$

By performing the procedure above, we have for each promoter $m$ and for each GO term $t$ assigned to it a value $P_t(e_t|S_m)$ that expresses how organized the space around $m$ is with respect to the GO term $t$. Figure 7 shows a contour plot of $P_t(e_t|S)$ that interpolates the values $P_t(e_t|S_m)$ calculated for all promoters $m$ with a GO term $t$=GO:0000003 "Reproduction" when mapped to the PCA-derived 2-dimensional space. Dark regions indicate high level of enrichment. The star marker highlights the promoter NM-002784 with the highest level of enrichment for the GO term GO:0000003. The plot reveals islands of highly enriched promoter regions related to a particular biological function.

To further investigate the enrichment of biologically meaningful information we performed a motif discovery on the sets $S_m$. As an example, consider a reference promoter NM-002784 (highly enriched for GO:0000003). $S_m$ contains 38 promoter sequences that were passed to MEME [35]. Several statistically significant motifs were found. The most significant 12-nucleotide motif was then passed to Tomtom [36] that (reassuringly) recognized the motif as the binding site of transcription factor Sfpi1 ($p = 0.0003$, $q = 0.23$, UniPROBE database). We stress that the main purpose of this experiment was to illustrate the potential of GMLPCA to take stock of more complicated patterns than zero-order compositional biases. A more involved application of our binary tensor decomposition in specific analysis of biological sequences (genomic or aminoacid) is a matter for our future research.

### 6.3 Discussion

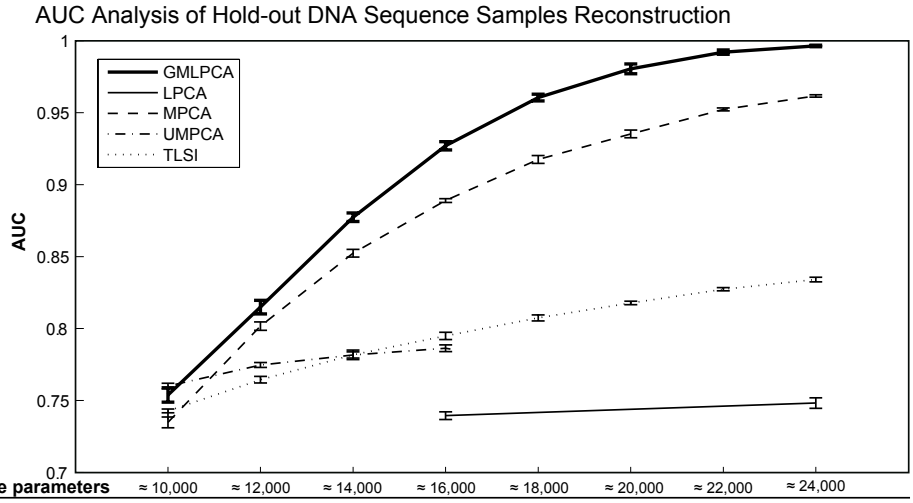
The experiments presented in this section compared our proposed GMLPCA model with other existing decomposition models in two different scenarios: (1) synthetic data compression and (2) topographic mapping of DNA sequences.

Experiments involving synthetic data sets generated from the linear Bernoulli natural parameter subspaces showed that GMLPCA clearly outperformed the other examined decomposition models. To investigate whether this translates to the case of real-world data, we repeated the reconstruction experiment on a data set of 2nd-order binary tensors representing DNA sequences used in section 6.2.

From the dataset of 62,000 sequences we randomly sampled 5 groups, each having 4,500 sequences. Each group was partitioned into 3,000 training and 1,500 test tensors. The tensor decomposition models were used on the training set to find latent subspaces spanned by different number of basis tensors. Then the hold-out sequences were projected onto the reduced-dimensionality latent space and subsequently reconstructed back into the original tensor space to measure the discrepancy between the reconstructed and original data. The procedure is identical to the one used in the synthetic data experiment.

Reconstruction results in terms of AUC for different latent space sizes are shown in figure 8. Our proposed GMLPCA model clearly outperformed other decomposition models except for the smallest latent subspace, where UMPCA model achieved slightly higher accuracy in terms of AUC.

Please note that we derived an iterative estimation scheme via maximum likelihood for fitting the GMLPCA model parameters. It is well known that good parameter initialization can be crucial for the success of maximum likelihood estimation. We empirically studied two commonly used initialization methods - random initialization and initialization by HOSVD (see e.g. [1]). The effects of these initialization methods on the convergence of log-likelihood were tested on both the synthetic data and DNA sequences. The results are summarized in figure 9. For the random initialization the average of 5 repeated runs is shown. It can be seen




| Number of free parameters | ≈ 10,000 | ≈ 12,000 | ≈ 14,000 | ≈ 16,000 | ≈ 18,000 | ≈ 20,000 | ≈ 22,000 | ≈ 24,000 |
|---|---|---|---|---|---|---|---|---|
| GMLPCA (row x column vectors) | 1 x 8 | 2 x 16 | 3 x 24 | 4 x 32 | 5 x 40 | 6 x 48 | 7 x 56 | 8 x 64 |
| MPCA (row x column vectors) | 1 x 8 | 2 x 16 | 3 x 24 | 4 x 32 | 5 x 40 | 6 x 48 | 7 x 56 | 8 x 64 |
| UMPCA (uncorrelated features) | 7 | 15 | 22 | 29 | - | - | - | - |
| TensorLSI (basis tensors) | 7 | 15 | 22 | 29 | 37 | 45 | 53 | 61 |
| LPCA (basis vectors) | - | - | - | 1 | - | - | - | 2 |

**Fig. 8** AUC analysis of hold-out 2nd-order binary tensor reconstructions obtained by the models using different number of free parameters among 5 disjoin subsets of binary tensors that represent DNA sequences. Table under the plot describes model settings for particular number of free parameters.

that for smaller subspaces ($R = [3 \times 3]$[3] for synthetic data and $R = [1 \times 8]$ for DNA sequences) the differences between the two initialization methods are negligible and GMLPCA converges in about 5 iterations. However, for larger subspaces the HOSVD initialization performed better than the blind random initialization and the training takes longer (GMLPCA converges in 20-25 iterations). Hence, the HOSVD initialization is superior and has been used in all our experiments.

We have also investigated the ability of tensor based models for unsupervised analysis and visualization of DNA sub-sequences (represented as binary tensors) from different functional regions based on the local term composition. By visualizing sub-sequence distributions in the principal sub-spaces spanned by the basis tensors, it transpires that the separation between promoters and introns is markedly better under the GMLPCA model than under the real-valued MPCA and TensorLSI methods. After detailed analysis, the discriminatory trends were identified as one of the most known signals in the promoter analysis domain verified by in-vivo biological experiments. To further investigate the method's utility for processing biological sequences, we searched the compressed feature space of promoters for biologically relevant structure. After assigning biologically meaningful labels to analyzed promoters, we found 75 GO terms that were enriched around one or more promoters. The observation that a subset of promoter sequences are functionally organized adds support to the method's ability to detect variation at an information-rich level.

[3] $R = [3 \times 3]$ represents a natural parameter subspace spanned by 3 row and 3 column vectors.

## 7 Semi-Supervised Extension

So far we considered the GMLPCA model as an unsupervised dimensionality reduction method for binary tensor data. However, many problems in machine learning involve decompositions that to certain degree preserve the label information provided for some data items. Such semi-supervised decomposition methods aim to benefit from both labeled and unlabeled data.

Here we propose to extend our GMLPCA model to the semi-supervised setting by forcing the model to search for a natural parameter subspace that represents a user specified compromise between the modelling quality and the degree of class separation. We do so by extending the cost function with a measure of separability of projected classes.

To enforce class separability of data items living in a metric space, Globerson and Roweis introduce a distribution over data items $l$, given a single data point $m$ [37]:

$$p(l|m) = \frac{e^{-d(m,l)}}{\sum_{k \neq m} e^{-d(m,k)}} \qquad m \neq l, \tag{23}$$

where $d(m, l)$ is the distance between the points $m$ and $l$. Loosely speaking, given a particular data item $m$, un-

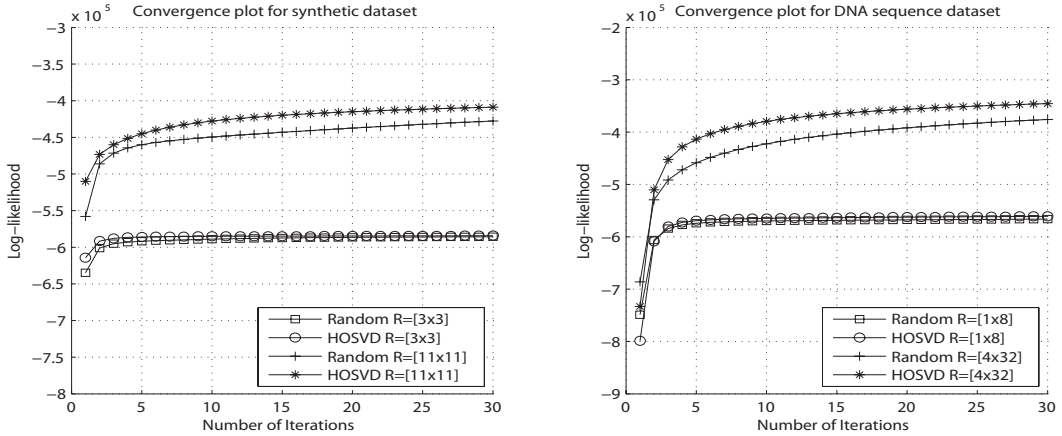



**Fig. 9** Illustration of the effect of initialization and convergence of GMLPCA: the evolution of log-likelihood on synthetic data (left plot) and DNA sequence dataset (right plot) over 30 iterations.

der $p(l|m)$ we are more likely to pick data points closer to $m$ than the more distant ones. In the ideal situation, where all points in the same class are collapsed to a single point and infinitely far from points of different classes, the conditional distributions (23) would become "bi-level" distributions [37]:

$$p_0(l|m) \propto \begin{cases} 1 & y_m = y_l \\ 0 & y_m \neq y_l, \end{cases} \tag{24}$$

where $y_m$ denotes a class label of data point $m$. In [37], maximal class separation under a given data model is achieved by tuning the model parameters so that the class divergence, $\sum_m \mathrm{KL}[p_0(\cdot|m)||p(\cdot|m)]$, is minimized. Minimizing $\sum_m \mathrm{KL}[p_0(\cdot|m)||p(\cdot|m)]$ is equivalent to maximizing

$$\sum_m \sum_{\substack{l:y_l=y_m \\ l\neq m}} \log\ p(l|m) = \tag{25}$$

$$\sum_m \frac{1}{c(y_m)-1} \sum_{\substack{l:y_l=y_m \\ l\neq m}} -d(l,m) - \log \sum_{\substack{k \\ k\neq m}} e^{-d(k,m)}, \tag{26}$$

where $c(y_m)$ denotes a number of points in class $y_m$.

Any two natural parameter tensors $\theta_m$ and $\theta_l$ living in the tensor subspace represent tensors of Bernoulli distributions $P(\mathcal{A}_m|\theta_m)$ and $P(\mathcal{A}_l|\theta_l)$ given by (6). The distance between those Bernoulli tensors is quantified by the symmetric KL divergence $D(m,l)$ (eqs. (21-22)). Using $D(m,l)$ as a metric on the subspace of tensors of Bernoulli natural parameters, (23) becomes

$$p(l|m) = \frac{e^{-D(m,l)}}{\sum_{k\neq m} e^{-D(m,k)}} \qquad m \neq l. \tag{27}$$

Given a subset of data tensors $\mathcal{D}_\ell \subset \mathcal{D}$ with class labels, the degree of projected class separation is quantified by (see (26))

$$\mathcal{F}(\mathcal{D}_\ell, \mathbf{y}) = \sum_{m\in\mathcal{D}_\ell} \frac{1}{c(y_m)-1} \times \sum_{\substack{l:y_l=y_m \\ l\neq m}} -D(l,m) - \log \sum_{\substack{k\in\mathcal{D}_\ell \\ k\neq m}} e^{-D(k,m)}, \tag{28}$$

where $\mathbf{y}$ is an $|\mathcal{D}_\ell|$-dimensional vector that contains labels for each data tensor in $\mathcal{D}_\ell$.

We aim to find tensor basis that simultaneously maximizes log-likelihood (7) of all training tensors and the degree of projected class separation (28): $\mathcal{L}(\mathcal{D}) + \beta\ \mathcal{F}(\mathcal{D}_\ell, \mathbf{y})$, where $\beta > 0$ is a regularization constant controlling the trade-off between data representation and separation.

To fit tensor basis, any optimization technique can be used. We used gradient ascent:

$$\mathbf{u}_{q,j}^{(n)} \leftarrow \mathbf{u}_{q,j}^{(n)} + \eta \left( \frac{\partial \mathcal{L}(\mathcal{D})}{\partial \mathbf{u}_{q,j}^{(n)}} + \beta \frac{\partial \mathcal{F}(\mathcal{D}_\ell, \mathbf{y})}{\partial \mathbf{u}_{q,j}^{(n)}} \right), \tag{29}$$

$$\Delta_{\mathbf{j}} \leftarrow \Delta_{\mathbf{j}} + \eta \left( \frac{\partial \mathcal{L}(\mathcal{D})}{\partial \Delta_{\mathbf{j}}} + \beta \frac{\partial \mathcal{F}(\mathcal{D}_\ell, \mathbf{y})}{\partial \Delta_{\mathbf{j}}} \right). \tag{30}$$

After each update cycle through the training set (updates of the projection space), the expansion coefficients (projections) of data tensors were calculated as described in section 5.

To illustrate workings of the semi-supervised tensor basis selection, we randomly sampled 500 DNA subsequences (250 introns and 250 promoters) as a training set and another set of 6,000 sub-sequences (3,000 introns + 3,000 promoters) as a hold-out set to verify whether the subspace found using the training set represents any global trends in the data. The model setting

(number of basis vectors for each mode) was exactly the same as in the previous experiment. The results are presented in figure 10. Plots in the first and second columns correspond to the training and hold-out sets, respectively. The first row represents a model with randomly chosen basis vectors for each mode. The second row corresponds to the completely unsupervised setting ($\beta = 0$). The third and fourth rows represent two different settings of class separation enforcement, $\beta = 5$ and $\beta = 20$, respectively. There is certain degree of natural class separation visible in the tensor subspace found in an unsupervised manner, without using any class label information. Random subspace position completely fails to discriminate between the two classes. However, further imposition of pressure for more class separation yields tensor basis giving only minute improvement in the class distribution over the completely unsupervised case. We conclude that the data is naturally split into two overlapping classes, that under the given subspace dimension, cannot be further separated into clearly distinct class projections. Biologically speaking, on the sequential level, many promoters and introns can exhibit similar subsequence structure.

## 8 Conclusion

This paper has introduced a generalized multilinear principal component analysis for binary tensors GMLPCA. The model can be considered as a generalization of the binary vector-based decomposition technique LPCA [11] to process binary tensors of an arbitrary order. Even though the original vector model is non-linear in parameters, the strong *linear* algebraic structure of the Tucker model for tensor decomposition can be superimposed on the parameter space of the tensor model, so that the efficient linear nature of parameter updates of [11] can be preserved.

Our experimental results involving synthetic and DNA sequence datasets showed that GMLPCA model is better suited to reconstruction of binary tensors than other examined decomposition models. Besides the reconstructions, we have also investigated the ability of tensor based models for unsupervised analysis and visualization of DNA sequences. By visualizing sequence distributions in the principal sub-spaces, it transpires that the separation between two functional classes of sequences is markedly better under GMLPCA model that under other used models.

In addition, we extended our GMLPCA model to the semi-supervised setting by forcing the model to search for a natural parameter subspace that represents a user specified compromise between the modelling quality and the degree of class separation. We used the semi-supervised setting of our model to further analyze the DNA sequences. However, implying a combined pressure for modelling quality and class separation of sequences yielded tensor basis giving only minute improvement in the class distribution over the completely unsupervised case. We conclude that in the tensor subspace discovered in the unsupervised setting, the data is already naturally split into two overlapping classes and cannot be further separated into more clearly distinct class projections by applying any additional supervised pressure.

## A Parameter Estimation

To get analytical parameter updates, we use the trick of [11] and take advantage of the fact that while the model log-likelihood (7) is not convex in the parameters, it is convex in any parameter, if the others are kept fixed. This leads to an iterative estimation scheme detailed below.

The analytical updates will be derived from a lower bound on the log-likelihood (7) using [11]:

$$\log\sigma(\hat{\theta}) \geq -\log 2 + \frac{\hat{\theta}}{2} - \log\cosh\left(\frac{\theta}{2}\right) - (\hat{\theta}^2 - \theta^2)\ \frac{\tanh\frac{\theta}{2}}{4\theta}, \quad (31)$$

where $\theta$ stands for the current value of individual natural parameters $\theta_{m,\mathbf{i}}$ of the Bernoulli noise models $P(\mathcal{A}_{m,\mathbf{i}}|\theta_{m,\mathbf{i}})$ and $\hat{\theta}$ stands for the future estimate of the parameter, given the current parameter values. Hence, from (7) we obtain[4]

$$\begin{aligned}
\mathcal{L}(\hat{\Theta}) &= \sum_{m=1}^{M}\sum_{\mathbf{i}\in\mathcal{Y}} \mathcal{A}_{m,\mathbf{i}}\log\sigma(\hat{\theta}_{m,\mathbf{i}}) + (1-\mathcal{A}_{m,\mathbf{i}})\log\sigma(-\hat{\theta}_{m,\mathbf{i}}) \\
&\geq \sum_{m=1}^{M}\sum_{\mathbf{i}\in\mathcal{Y}} \mathcal{A}_{m,\mathbf{i}}\Bigg[-\log 2 + \frac{\hat{\theta}_{m,\mathbf{i}}}{2} \\
&\qquad - \log\cosh\left(\frac{\theta_{m,\mathbf{i}}}{2}\right) - (\hat{\theta}^2_{m,\mathbf{i}} - \theta^2_{m,\mathbf{i}})\frac{\tanh\frac{\theta_{m,\mathbf{i}}}{2}}{4\theta_{m,\mathbf{i}}}\Bigg] \\
&\quad + (1-\mathcal{A}_{m,\mathbf{i}})\Bigg[-\log 2 - \frac{\hat{\theta}_{m,\mathbf{i}}}{2} \\
&\qquad - \log\cosh\left(\frac{\theta_{m,\mathbf{i}}}{2}\right) - (\hat{\theta}^2_{m,\mathbf{i}} - \theta^2_{m,\mathbf{i}})\frac{\tanh\frac{\theta_{m,\mathbf{i}}}{2}}{4\theta_{m,\mathbf{i}}}\Bigg] \qquad (32) \\
&= H(\hat{\Theta},\Theta). \qquad (33)
\end{aligned}$$

Denote $(\tanh\frac{\theta_{m,\mathbf{i}}}{2})/\theta_{m,\mathbf{i}}$ by $\Psi_{m,\mathbf{i}}$. Grouping together constant terms in (32) leads to

$$\begin{aligned}
H(\hat{\Theta},\Theta) = \sum_{m=1}^{M}\sum_{\mathbf{i}\in\mathcal{Y}} &\left[\hat{\theta}_{m,\mathbf{i}}\left(\mathcal{A}_{m,\mathbf{i}} - \frac{1}{2}\right) - \frac{\Psi_{m,\mathbf{i}}}{4}\ \hat{\theta}^2_{m,\mathbf{i}}\right] \\
&+ \mathit{Const.} \qquad (34)
\end{aligned}$$

Note that $H(\hat{\Theta},\Theta) = \mathcal{L}(\hat{\Theta})$ only if $\hat{\Theta} = \Theta$. Therefore by choosing $\hat{\Theta}$ that maximizes $H(\hat{\Theta},\Theta)$ we guarantee $\mathcal{L}(\hat{\Theta}) \geq H(\hat{\Theta},\Theta) \geq H(\Theta,\Theta) = \mathcal{L}(\Theta)$ [11].

[4] $\theta$'s are fixed current values of the parameters and should be treated as constants

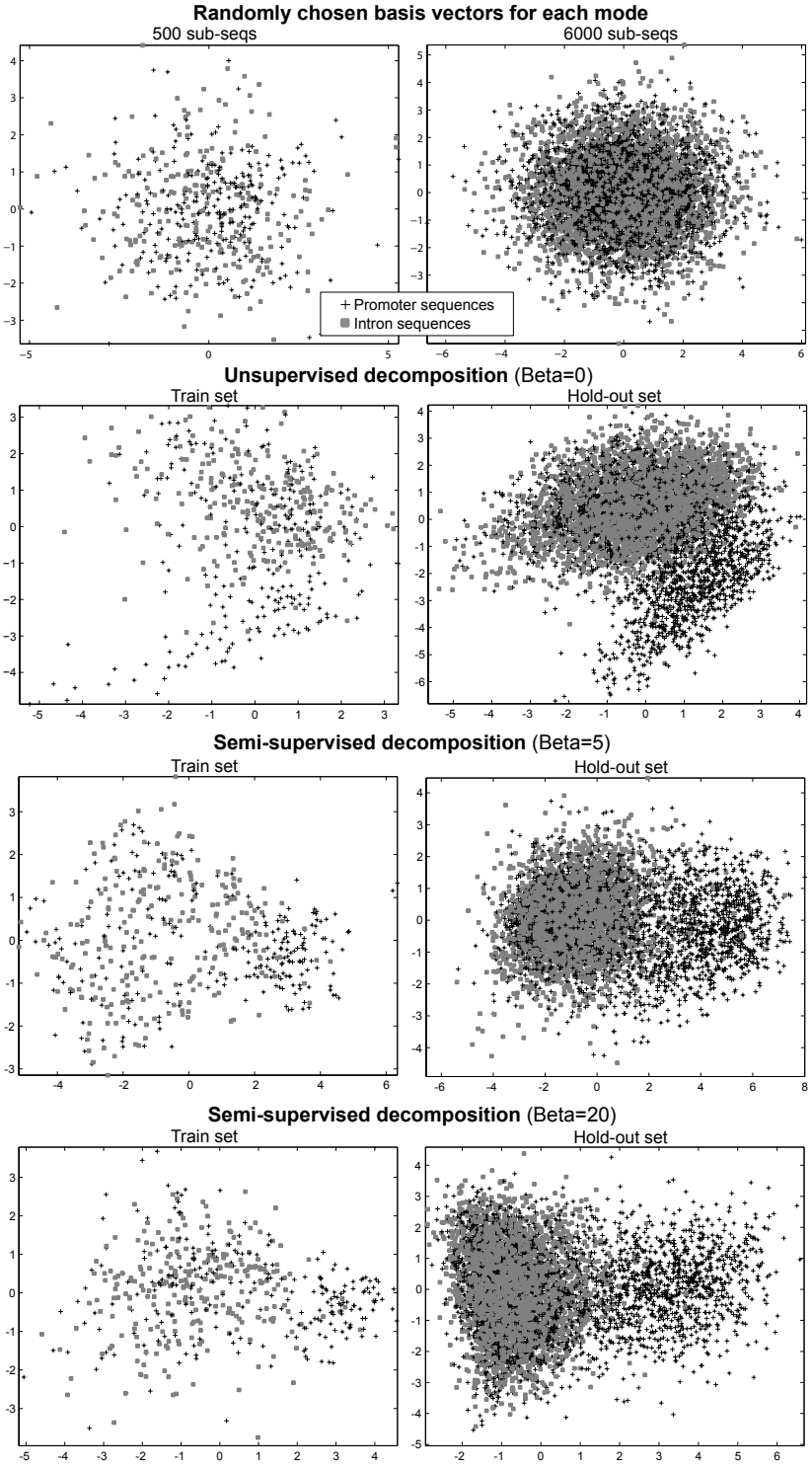



**Fig. 10** Two-dimensional PCA projections from 4 different tensor spaces of training and hold-out sets of 500 and 6,000 randomly sampled sub-sequences, respectively. The first row represents a tensor model with randomly chosen basis vectors for each mode. The second row corresponds to the completely unsupervised setting ($\beta = 0$). The third and fourth rows represent two different settings of class separation enforcement, $\beta = 5$ and $\beta = 20$, respectively.

We are now ready to constrain the Bernoulli parameters to be optimized (see (9)):

$$\hat{\theta}_{m,\mathbf{i}} = \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}} \cdot \prod_{n=1}^{N} u^{(n)}_{r_n,i_n} + \Delta_{\mathbf{i}}. \tag{35}$$

We will update the model parameters so as to maximize

$$\mathcal{H} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}} \mathcal{H}_{m,\mathbf{i}}, \tag{36}$$

where

$$\mathcal{H}_{m,\mathbf{i}} = \left(\mathcal{A}_{m,\mathbf{i}} - \frac{1}{2}\right)\hat{\theta}_{m,\mathbf{i}} - \frac{\Psi_{m,\mathbf{i}}}{4}\ \hat{\theta}^2_{m,\mathbf{i}}, \tag{37}$$

with $\hat{\theta}_{m,\mathbf{i}}$ given by (35).

## A.1 Updates for $n$-mode space basis

When updating the $n$-mode space basis $\{\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{R_n}\}$, the bias tensor $\Delta$ and the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, $m = 1, 2, ...M$, $\mathbf{r} \in \rho$, are kept fixed to their current values.

For $n = 1, 2, ..., N$, define

$$\begin{aligned}\mathcal{Y}_{-n} =& \{1, 2, ..., I_1\} \times ... \times \{1, 2, ..., I_{n-1}\} \times \{1\} \\ & \times \{1, 2, ..., I_{n+1}\} \times ... \times \{1, 2, ..., I_N\},\end{aligned} \tag{38}$$

with obvious interpretation in the boundary cases. Given $\mathbf{i} \in \mathcal{Y}_{-n}$ and an $n$-mode index $j \in \{1, 2, ..., I_n\}$, the index $N$-tuple $(i_1, ..., i_{n-1}, j, i_{n+1}, ..., i_N)$ formed by inserting $j$ at the $n$th place of $\mathbf{i}$ is denoted by $[\mathbf{i}, j|n]$.

In order to evaluate

$$\frac{\partial\ \mathcal{H}}{\partial\ u^{(n)}_{q,j}}, \quad q = 1, 2, ..., R_n,\ j = 1, 2, ..., I_n,$$

we realize that $u^{(n)}_{q,j}$ is involved in expressing all $\hat{\theta}_{m,[\mathbf{i},j|n]}$, $m = 1, 2, ..., M$, with $\mathbf{i} \in \mathcal{Y}_{-n}$. Therefore,

$$\frac{\partial\ \mathcal{H}}{\partial\ u^{(n)}_{q,j}} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}_{-n}} \frac{\partial\ \mathcal{H}_{m,[\mathbf{i},j|n]}}{\partial\ \hat{\theta}_{m,[\mathbf{i},j|n]}} \frac{\partial\ \hat{\theta}_{m,[\mathbf{i},j|n]}}{\partial\ u^{(n)}_{q,j}}, \tag{39}$$

where

$$\frac{\partial\ \mathcal{H}_{m,[\mathbf{i},j|n]}}{\partial\ \hat{\theta}_{m,[\mathbf{i},j|n]}} = \left(\mathcal{A}_{m,[\mathbf{i},j|n]} - \frac{1}{2}\right) - \frac{\Psi_{m,[\mathbf{i},j|n]}}{2}\ \hat{\theta}_{m,[\mathbf{i},j|n]} \tag{40}$$

and from (35),

$$\frac{\partial\ \hat{\theta}_{m,[\mathbf{i},j|n]}}{\partial\ u^{(n)}_{q,j}} = \mathcal{B}^{(n)}_{m,\mathbf{i},q} = \sum_{\mathbf{r}\in\rho_{-n}} \mathcal{Q}_{m,[\mathbf{r},q|n]} \cdot \prod_{s=1,s\neq n}^{N} u^{(s)}_{r_s,i_s}. \tag{41}$$

Here, the index set $\rho_{-n}$ is defined analogously to $\mathcal{Y}_{-n}$:

$$\begin{aligned}\rho_{-n} =& \{1, 2, ..., R_1\} \times ... \times \{1, 2, ..., R_{n-1}\} \times \{1\} \\ & \times \{1, 2, ..., R_{n+1}\} \times ... \times \{1, 2, ..., R_N\}.\end{aligned} \tag{42}$$

Setting the derivative (39) to zero results in

$$\begin{aligned}\sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}_{-n}} (2\mathcal{A}_{m,[\mathbf{i},j|n]} - 1)\ \mathcal{B}^{(n)}_{m,\mathbf{i},q} =& \\ \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}_{-n}} \Psi_{m,[\mathbf{i},j|n]}\ \hat{\theta}_{m,[\mathbf{i},j|n]}\ \mathcal{B}^{(n)}_{m,\mathbf{i},q}.&\end{aligned} \tag{43}$$

Rewriting (35) as

$$\begin{aligned}\hat{\theta}_{m,[\mathbf{i},j|n]} = \sum_{t=1}^{R_n} \sum_{\mathbf{r}\in\rho_{-n}} \mathcal{Q}_{m,[\mathbf{r},t|n]}\ u^{(n)}_{t,j} \prod_{s=1,s\neq n}^{N} u^{(s)}_{r_s,i_s} \\ +\ \Delta_{[\mathbf{i},j|n]}\end{aligned} \tag{44}$$

and applying to (43) we obtain

$$\sum_{t=1}^{R_n} u^{(n)}_{t,j}\ \mathcal{K}^{(n)}_{q,t,j} = \mathcal{S}^{(n)}_{q,j}, \tag{45}$$

where

$$\mathcal{S}^{(n)}_{q,j} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}_{-n}} (2\mathcal{A}_{m,[\mathbf{i},j|n]} - 1 - \Psi_{m,[\mathbf{i},j|n]}\Delta_{[\mathbf{i},j|n]})\mathcal{B}^{(n)}_{m,\mathbf{i},q}, \tag{46}$$

and

$$\begin{aligned}\mathcal{K}^{(n)}_{q,t,j} = \sum_{m=1}^{M} \sum_{\mathbf{r}\in\rho_{-n}} \mathcal{Q}_{m,[\mathbf{r},t|n]} \\ \times \sum_{\mathbf{i}\in\mathcal{Y}_{-n}} \Psi_{m,[\mathbf{i},j|n]}\ \mathcal{B}^{(n)}_{m,\mathbf{i},q} \prod_{s=1,s\neq n}^{N} u^{(s)}_{r_s,i_s}.\end{aligned} \tag{47}$$

For each $n$-mode coordinate $j \in \{1, 2, ..., I_n\}$, collect the $j$-th coordinate values of all $n$-mode basis vectors into a column vector $\mathbf{u}^{(n)}_{:,j} = (u^{(n)}_{1,j}, u^{(n)}_{2,j}, ..., u^{(n)}_{R_n,j})^T$. Analogously, stack all the $\mathcal{S}^{(n)}_{q,j}$ values in a column vector $\mathcal{S}^{(n)}_{:,j} = (\mathcal{S}^{(n)}_{1,j}, \mathcal{S}^{(n)}_{2,j}, ..., \mathcal{S}^{(n)}_{R_n,j})^T$. Finally, we construct an $R_n \times R_n$ matrix $\mathcal{K}^{(n)}_{:,:,j}$ whose $q$-th row is $(\mathcal{K}^{(n)}_{q,1,j}, \mathcal{K}^{(n)}_{q,2,j}, ..., \mathcal{K}^{(n)}_{q,R_n,j})$, $q = 1, 2, ..., R_n$. The $n$-mode basis vectors are updated by solving $I_n$ linear systems of size $R_n \times R_n$:

$$\mathcal{K}^{(n)}_{:,:,j}\ \mathbf{u}^{(n)}_{:,j} = \mathcal{S}^{(n)}_{:,j}, \quad j = 1, 2, ..., I_n. \tag{48}$$

## A.2 Updates for expansion coefficients

When updating the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, the bias tensor $\Delta$ and the basis sets $\{\mathbf{u}^{(n)}_1, \mathbf{u}^{(n)}_2, ..., \mathbf{u}^{(n)}_{R_n}\}$ for all $n$ modes $n = 1, 2, ..., N$ are kept fixed to their current values.

For $\mathbf{r} \in \rho$ and $\mathbf{i} \in \mathcal{Y}$ denote $\prod_{n=1}^{N} u^{(n)}_{r_n,i_n}$ by $C_{\mathbf{r},\mathbf{i}}$. For data index $\ell = 1, 2, ..., M$ and basis index $\mathbf{v} \in \rho$ we have

$$\frac{\partial\ \mathcal{H}}{\partial\ \mathcal{Q}_{\ell,\mathbf{v}}} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}} \frac{\partial\ \mathcal{H}_{m,\mathbf{i}}}{\partial\ \hat{\theta}_{m,\mathbf{i}}} \frac{\partial\ \hat{\theta}_{m,\mathbf{i}}}{\partial\ \mathcal{Q}_{\ell,\mathbf{v}}}, \tag{49}$$

where

$$\frac{\partial\ \mathcal{H}_{m,\mathbf{i}}}{\partial\ \hat{\theta}_{m,\mathbf{i}}} = \left(\mathcal{A}_{m,\mathbf{i}} - \frac{1}{2}\right) - \frac{\Psi_{m,\mathbf{i}}}{2}\ \hat{\theta}_{m,\mathbf{i}} \tag{50}$$

and $\frac{\partial\ \hat{\theta}_{m,\mathbf{i}}}{\partial\ \mathcal{Q}_{\ell,\mathbf{v}}} = C_{\mathbf{v},\mathbf{i}}$ if $m = \ell$ and $\frac{\partial\ \hat{\theta}_{m,\mathbf{i}}}{\partial\ \mathcal{Q}_{\ell,\mathbf{v}}} = 0$ otherwise.

By imposing $\frac{\partial\ \mathcal{H}}{\partial\ \mathcal{Q}_{\ell,\mathbf{v}}} = 0$, we get

$$\mathcal{T}_{\mathbf{v},\ell} = \sum_{\mathbf{r}\in\rho} \mathcal{P}_{\mathbf{v},\mathbf{r},\ell}\ \mathcal{Q}_{\ell,\mathbf{r}}, \tag{51}$$

where

$$\mathcal{T}_{\mathbf{v},\ell} = \sum_{\mathbf{i}\in\mathcal{Y}} (2\mathcal{A}_{\ell,\mathbf{i}} - 1 - \Psi_{\ell,\mathbf{i}}\ \Delta_{\mathbf{i}})\ C_{\mathbf{v},\mathbf{i}} \tag{52}$$

and

$$\mathcal{P}_{\mathbf{v},\mathbf{r},\ell} = \sum_{\mathbf{i}\in\mathcal{Y}} \Psi_{\ell,\mathbf{i}}\ C_{\mathbf{v},\mathbf{i}}\ C_{\mathbf{r},\mathbf{i}}. \tag{53}$$

To solve for expansion coefficients using the tools of matrix algebra, we need to vectorize tensor indices. Consider any one-to-one function $\kappa$ from $\rho$ to $\{1, 2, ..., \prod_{n=1}^{N} R_n\}$. For each input tensor index $\ell = 1, 2, ..., M$,

- create a square $(\prod_{n=1}^{N} R_n)\times(\prod_{n=1}^{N} R_n)$ matrix $\mathcal{P}_{:,:,\ell}$ whose $(\kappa(\mathbf{v}), \kappa(\mathbf{r}))$-th element is equal to $\mathcal{P}_{\mathbf{v},\mathbf{r},\ell}$,
- stack the values of $\mathcal{T}_{\mathbf{v},\ell}$ into a column vector $\mathcal{T}_{:,\ell}$ whose $\kappa(\mathbf{v})$-th coordinate is $\mathcal{T}_{\mathbf{v},\ell}$,
- collect the expansion coefficients $\mathcal{Q}_{\ell,\mathbf{r}}$ in a column vector $\mathcal{Q}_{\ell,:}$ with $\kappa(\mathbf{r})$-th coordinate equal to $\mathcal{Q}_{\ell,\mathbf{r}}$.

The expansion coefficients for the $\ell$-th input tensor $\mathcal{A}_\ell$ can be obtained by solving

$$\mathcal{P}_{:,:,\ell}\ \mathcal{Q}_{\ell,:} = \mathcal{T}_{:,\ell}, \quad \ell = 1, 2, ..., M. \tag{54}$$

### A.3 Updates for the bias tensor

As before, when updating the bias tensor $\Delta$, the expansion coefficients $\mathcal{Q}_{m,\mathbf{r}}$, $m = 1, 2, ...M$, $\mathbf{r} \in \rho$, and the basis sets $\{\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, ..., \mathbf{u}_{R_n}^{(n)}\}$ for all $n$ modes $n = 1, 2, ..., N$ are kept fixed to their current values.

Fix $\mathbf{j} \in \mathcal{Y}$. We evaluate

$$\frac{\partial\ \mathcal{H}}{\partial\ \Delta_{\mathbf{j}}} = \sum_{m=1}^{M} \sum_{\mathbf{i}\in\mathcal{Y}} \frac{\partial\ \mathcal{H}_{m,\mathbf{i}}}{\partial\ \hat{\theta}_{m,\mathbf{i}}}\ \frac{\partial\ \hat{\theta}_{m,\mathbf{i}}}{\partial\ \Delta_{\mathbf{j}}}, \tag{55}$$

where $\frac{\partial\ \hat{\theta}_{m,\mathbf{i}}}{\partial\ \Delta_{\mathbf{j}}}$ is equal to 1 if $\mathbf{i} = \mathbf{j}$ and 0 otherwise.

Solving for $\frac{\partial\ \mathcal{H}}{\partial\ \Delta_{\mathbf{j}}} = 0$ leads to

$$\Delta_{\mathbf{j}} = \frac{\sum_{m=1}^{M} 2\mathcal{A}_{m,\mathbf{j}} - 1 - \Psi_{m,\mathbf{j}} \cdot \sum_{\mathbf{r}\in\rho} \mathcal{Q}_{m,\mathbf{r}}\ C_{\mathbf{r},\mathbf{j}}}{\sum_{m=1}^{M} \Psi_{m,\mathbf{j}}}. \tag{56}$$

**Acknowledgements** Jakub Mažgut was supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11. Peter Tiňo was supported by the DfES UK/Hong Kong Fellowship for Excellence and a BBSRC grant (no. BB/H012508/1). Mikael Bodén was supported by the ARC Centre of Excellence in Bioinformatics and the 2009 University of Birmingham Ramsay Research Scholarship Award. Hong Yan is supported by a grant from City University of Hong Kong (Project 7002843).

## References

1. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, pp. 18–39, Jan. 2008.
2. C. Nolker and H. Ritter, "Visual recognition of continuous hand postures," *IEEE Trans. Neural Netw.*, vol. 13, pp. 983–994, July 2002.
3. K. Jia and S. Gong, "Multi-modal tensor face for simultaneous super-resolution and recognition," in *10th IEEE Int. Conf. Computer Vision*, vol. 2, (Beijing), pp. 1683–1690, Oct. 2005.
4. N. Renard and S. Bourennane, "An ICA-based multilinear algebra tools for dimensionality reduction in hyperspectral imagery," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, (Las Vegas, NV), pp. 1345–1348, Apr. 2008.
5. D. Cai, X. He, and J. Han, "Tensor space model for document analysis," in *Proc. 29th Annu. ACM SIGIR Int. Conf. Research and Development in Information Retrieval*, (Seatlle, WA), pp. 625–626, Aug. 2006.
6. K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.
7. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Trans. Neural Netw.*, vol. 20, pp. 103–123, Jan. 2009.
8. S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Trans. Neural Netw.*, vol. 20, pp. 217–235, Feb. 2009.
9. Y. Panagakis, C. Kotropoulos, and G. Arce, "Nonnegative multilinear principal component analysis of auditory temporal modulations for music genre classification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 576–588, March 2010.
10. J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, "Symmetric tensor decomposition," *Linear Algebra Appl.*, vol. In Press, Corrected Proof, 2010.
11. A. Schein, L. Saul, and L. Ungar, "A generalized linear model for principal component analysis of binary data," in *9th Int. Workshop Artificial Intelligence and Statistics*, (Key West, FL), Jan. 2003.
12. E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Trans. Knowl. Data En.*, vol. 21, pp. 6 –20, Jan. 2009.
13. H. Wang and N. Ahuja, "Rank-r approximation of tensors: Using image-as-matrix representation," in *Computer Vision and Pattern Recognition*, pp. 346–353, 2005.
14. L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Applicat.*, vol. 21, no. 4, pp. 1253–1278, 2000.
15. E. Kofidis and P. A. Regalia, "On the best rank-1 approximation of higher-order supersymmetric tensors," *SIAM J. Matrix Anal. Applicat.*, vol. 23, no. 3, pp. 863–884, 2001.
16. H. Wang and N. Ahuja, "Compact representation of multidimensional data using tensor rank-one decomposition," in *Proc. 17th Int. Conf. Pattern Recognition*, (Cambridge, UK), pp. 44–47, Aug. 2004.
17. J. Ye, R. Janardan, and Q. Li, "Gpca: an efficient dimension reduction scheme for image compression and retrieval," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, (New York, NY, USA), pp. 354–363, ACM, 2004.
18. D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, pp. 36–47, Jan. 2008.

19. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning.," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1820–1836, 2009.
20. K. Inoue, K. Hara, and K. Urahama, "Robust multilinear principal component analysis," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 591–597, 29 2009-oct. 2 2009.
21. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recogn.*, vol. 44, pp. 1540–1551, July 2011.
22. C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Advances in Neural Information Processing Systems*, vol. 16, (Banff, AL, Canada), pp. 313–320, July 2003.
23. X. Li, J. Zeng, and H. Yan, "PCA-HPR: A principle component analysis model for human promoter recognition," *Bioinformation*, vol. 2, no. 9, pp. 373–378, 2008.
24. S. Sonnenburg, A. Zien, P. Philips, and G. Ratsch, "POIMs: positional oligomer importance matrices–understanding support vector machine-based signal detectors," *Bioinformatics*, vol. 24, no. 13, pp. i6–i14, 2008.
25. P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*. The MIT Press, 2 ed., August 2001.
26. A. Isaev, *Introduction to Mathematical Methods in Bioinformatics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.
27. H. Wakaguri, R. Yamashita, S. Y., S. Sugano, and K. Nakai, "DBTSS: Database of transcription start sites," *Nucleic Acids Res.*, vol. 36, no. Database-Issue, pp. 97–101, 2008.
28. S. Saxonov, I. Daizadeh, A. Fedorov, and W. Gilbert, "EID: The exon-intron database - an exhaustive database of protein-coding intron-containing genes," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 185–190, 2000.
29. D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: learning probabilistic automata with variable memory length," *Mach. Learning*, vol. 25, no. 2-3, pp. 117–149, 1996.
30. P. Tino and G. Dorffner, "Predicting the future of discrete sequences from fractal representations of the past," *Mach. Learning*, vol. 45, no. 2, pp. 187–217, 2001.
31. S. Cross, V. Clark, and A. Bird, "Isolation of CpG islands from large genomic clones," *Nucleic Acids Res.*, vol. 27, no. 10, pp. 2099–2107, 1999.
32. M. Bodén and T. L. Bailey, "Associating transcription factor-binding site motifs with target GO terms and target genes.," *Nucleic Acids Res.*, vol. 36, no. 12, pp. 4108–4117, 2008.
33. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.
34. R. Straussman, D. Nejman, D. Roberts, I. Steinfeld, B. Blum, N. Benvenisty, I. Simon, Z. Yakhini, and H. Cedar, "Developmental programming of CpG island methylation profiles in the human genome," *Nature Structural & Molecular Biology*, vol. 16, no. 5, pp. 564–571, 2009.
35. T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble, "Meme suite: tools for motif discovery and searching.," *Nucleic Acids Res.*, vol. 37, no. Web Server issue, pp. W202–208, 2009.
36. S. Gupta, J. Stamatoyannopoulos, T. Bailey, and W. Noble, "Quantifying similarity between motifs," *Genome Biology*, vol. 8, no. 2, p. R24, 2007.
37. A. Globerson and S. Roweis, "Metric learning by collapsing classes," *Advances in Neural Information Processing Systems*, vol. 18, pp. 451–458, 2006.