

# Generalised Coupled Tensor Factorisation

**Authored by:**

Kenan Y. Yılmaz

Ali T. Cemgil

Umut Simsekli

## **Abstract**

We derive algorithms for generalised tensor factorisation (GTF) by building upon the well-established theory of Generalised Linear Models. Our algorithms are general in the sense that we can compute arbitrary factorisations in a message passing framework, derived for a broad class of exponential family distributions including special cases such as Tweedie's distributions corresponding to  $\beta$ -divergences. By bounding the step size of the Fisher Scoring iteration of the GLM, we obtain general updates for real data and multiplicative updates for non-negative data. The GTF framework is, then extended easily to address the problems when multiple observed tensors are factorised simultaneously. We illustrate our coupled factorisation approach on synthetic data as well as on a musical audio restoration problem.

## **1 Paper Body**

A fruitful modelling approach for extracting meaningful information from highly structured multivariate datasets is based on matrix factorisations (MFs). In fact, many standard data processing methods of machine learning and statistics such as clustering, source separation, independent components analysis (ICA), nonnegative matrix factorisation (NMF), latent semantic indexing (LSI) can be expressed and understood as MF problems. These MF models also have well understood probabilistic interpretations as probabilistic generative models. Indeed, many standard algorithms mentioned above can be derived as maximum likelihood or maximum a-posteriori parameter estimation procedures. It is also possible to do a full Bayesian treatment for model selection [1]. Tensors appear as a natural generalisation of matrix factorisation, when observed data and/or a latent representation have several semantically meaningful dimensions. Before giving a formal definition, consider the following motivating example  $X_{i,j,r}$   $X_{j,r,p}$   $X_{j,r,q}$   $X_{i,j,k}$   $Z_1 Z_2 Z_3 X_{2j,p}$   $Z_2 Z_4 X_{3j,q}$   $Z_2 Z_5$  (1)  $r$

$r$

where  $X_1$  is an observed 3-way array and  $X_2, X_3$  are 2-way arrays, while  $Z_\ell$  for  $\ell = 1 \dots 5$  are the latent 2-way arrays. Here, the 2-way arrays are just matrices but this can be easily extended to objects having arbitrary number of indices. As the term ‘N-way array’ is awkward, we prefer using the more convenient term tensor. Here,  $Z_2$  is a shared factor, coupling all models. As the first model is a CP (Parafac) while the second and the third ones are MF’s, we call the combined factorization as CP/MF/MF model. Such models are of interest when one can obtain different ‘views’ of the same piece of information (here  $Z_2$ ) under different experimental conditions. Singh and Gordon [2] focused on a similar problem called as collective matrix factorisation (CMF) or multi-matrix factorisation, for relational learning but only for matrix factors and observations. In addition, their generalised Bregman divergence minimisation procedure assumes matching link and loss functions. For coupled matrix and tensor factorization (CMTF), recently [3] proposed a gradient-based allat-once optimization method as an alternative to alternating least square (ALS) optimization and

1 demonstrated their approach for a CP/MF coupled model. Similar models are used for proteinprotein interactions (PPI) problems in gene regulation [4]. The main motivation of the current paper is to construct a general and practical framework for computation of tensor factorisations (TF), by extending the well-established theory of Generalised Linear Models (GLM). Our approach is also partially inspired by probabilistic graphical models: our computation procedures for a given factorisation have a natural message passing interpretation. This provides a structured and efficient approach that enables very easy development of application specific custom models, priors or error measures as well as algorithms for joint factorisations where an arbitrary set of tensors can be factorised simultaneously. Well known models of multiway analysis (Parafac, Tucker [5]) appear as special cases and novel models and associated inference algorithms can be automatically be developed. In [6], the authors take a similar approach to tensor factorisations as ours, but that work is limited to KL and Euclidean costs, generalising MF models of [7] to the tensor case. It is possible to generalise this line of work to  $\alpha$ -divergences [8] but none of these works address the coupled factorisation case and consider only a restricted class of cost functions.

2

## Generalised Linear Models for Matrix/Tensor Factorisation

To set the notation and our approach, we briefly review GLMs following closely the original notation of [9, ch 5]. A GLM assumes that a data vector  $x$  has conditionally independently drawn components  $x_i$  according to an exponential family density  $x \sim b(\eta)$

$$p(x_i) = \frac{1}{Z(\eta_i)} \exp(\eta_i x_i) \exp(-\log Z(\eta_i)) \quad \text{var}(x_i) = \eta_i^2 c(\eta_i, \eta_i) \quad \eta_i = x_i = (2)$$

Here,  $\eta_i$  are canonical parameters and  $\eta^2$  is a known dispersion parameter.  $\eta_i$  is the expectation of  $x_i$  and  $b(\eta)$  is the log partition function, enforcing normalization. The canonical parameters are not directly estimated, instead one assumes a link function  $g(\eta)$  that ‘links’ the mean of the distribution  $x_i$  and

assumes that  $g(\eta_i) = \eta_i^T z$  where  $\eta_i$  is the  $i$ th row vector of a known model matrix  $L$  and  $z$  is the parameter vector to be estimated,  $A^T$  denotes matrix transpose of  $A$ . The model is linear in the sense that a function of the mean is linear in parameters, i.e.,  $g(\eta_i) = Lz$ . A Linear Model (LM) is a special case of GLM that assumes normality, i.e.  $\eta_i \sim N(\eta_i; \mu_i, \sigma^2)$  as well as linearity that implies identity link function as  $g(\eta_i) = \mu_i = \eta_i^T z$  assuming  $\mu_i$  are known. Logistic regression assumes a log link,  $g(\eta_i) = \log \mu_i = \eta_i^T z$ ; here  $\log \mu_i$  and  $z$  have a linear relationship [9]. The goal in classical GLM is to estimate the parameter vector  $z$ . This is typically achieved via a Gauss-Newton method (Fisher Scoring). The necessary objects for this computation are the log likelihood, the derivative and the Fisher Information (the expected value of negative of the Fisher Score). These are easily derived as:  $X^T X^{-1} X^T L = [x_i^T b'(\eta_i)] / \sigma^2 = 2 \text{c}(x_i, \eta_i) (x_i^T x_i) w_i g(\eta_i) \eta_i$  (3)  $\eta_i^T z = \eta_i^T L^{-1} L^T L z = 2 L^T D G(x_i^T x_i) (4) 2 \eta_i^T z = \eta_i^T z$

where  $w$  is a vector with elements  $w_i$ ,  $D$  and  $G$  are the diagonal matrices as  $D = \text{diag}(w)$ ,  $G = \text{diag}(g(\eta_i))$  and

$\eta_i^T g(\eta_i) (5) w_i = v(\eta_i) g(\eta_i) g(\eta_i) = \eta_i^T \eta_i$  with  $v(\eta_i)$  being the variance function related to the observation variance by  $\text{var}(\eta_i) = 2 v(\eta_i)$ . Via Fisher Scoring, the general update equation in matrix form is written as

$$\eta_i^T z = z + L^T D L^{-1} D G(x_i^T x_i) (6)$$

Although this formulation is somewhat abstract, it covers a very broad range of model classes that are used in practice. For example, an important special case appears when the variance functions are in the form of  $v(\eta_i) = \eta_i^T p$ . By setting  $p = \{0, 1, 2, 3\}$  these correspond to Gaussian, Poisson, Exponential/Gamma, and Inverse Gaussian distributions [10, pp.30], which are special cases of the exponential family of distributions for any  $p$  named Tweedie's family [11]. Those for  $p = \{0, 1, 2\}$ , in turn, correspond to EU, KL and IS cost functions often used for NMF decompositions [12, 7].

## 2.1

Tensor Factorisations (TF) as GLMs

The key observation for expressing a TF model as a GLM is to identify the multilinear structure and using an alternating optimization approach. To hide the notational complexity, we will give an example with a simple matrix factorisation model; extension to tensors will require heavier notation, but are otherwise conceptually straightforward. Consider a MF model  $X_{i,j,r} = Z_1 Z_2^T \eta_{i,j} = g(X)$  in scalar  $g(X) = Z_1 Z_2^T (7) r$

$\eta_{i,j}$  are matrices of compatible sizes. Indeed, by applying the  $\text{vec}$  operator where  $Z_1$ ,  $Z_2$  and  $g(X)$  (vectorization, stacking columns of a matrix to obtain a vector) to both sides of (7) we obtain two equivalent representation of the same system  $\eta = (I - j - \eta^T Z_1) \text{vec}(Z_2) = \text{vec}(g(X))$

$$\eta^T g(X) (\eta^T Z_1 Z_2^T)^T \text{vec}(Z_2) = \text{vec}(Z_2)^T \eta^T Z_2^T Z_2^T Z_2^T$$

(8)

where  $I - j - \eta^T$  denotes the  $j - \eta^T$  identity matrix,  $\eta^T$  denotes the Kronecker product [13], and  $\text{vec} Z$  Clearly, this is a GLM where  $\eta^T$  plays the role of a model matrix and  $Z_2$  is the parameter  $Z$ . vector. By alternating

between  $Z_1$  and  $Z_2$ , we can maximise the log-likelihood iteratively; indeed this alternating maximisation is standard for solving matrix factorisation problems. In the sequel, we will show that a much broader range of algorithms can be readily derived in the GLM framework. 2.2

#### Generalised Tensor Factorisation

We define a tensor  $\mathcal{Z}$  as a multiway array with an index set  $V = \{i_1, i_2, \dots, i_N\}$  where each index  $i_n$  for  $n = 1 \dots N$  runs as  $i_n = 1 \dots I_n$ . An element of the tensor  $\mathcal{Z}$  is a scalar that we denote by  $\mathcal{Z}(i_1, i_2, \dots, i_N)$  or  $\mathcal{Z}_{i_1, i_2, \dots, i_N}$  or as a shorthand notation by  $\mathcal{Z}(\mathbf{v})$  with  $\mathbf{v}$  being a particular configuration.  $\mathbf{v}$  denotes number of all distinct configurations for  $V$ , and e.g. if  $V = \{i_1, i_2\}$  then  $\mathbf{v} = i_1 i_2$ . We call the form  $\mathcal{Z}(\mathbf{v})$  as element-wise; the notation  $[\mathcal{Z}]$  yields a tensor by enumerating all the indices, i.e.,  $\mathcal{Z} = [\mathcal{Z}_{i_1, i_2, \dots, i_N}]$  or  $\mathcal{Z} = [\mathcal{Z}(\mathbf{v})]$ . For any two tensors  $X$  and  $Y$  of compatible order,  $X \mathcal{Z} Y$  is an element-wise multiplication and if not explicitly stressed  $X/Y$  is an element-wise division.  $\mathbf{1}$  is an object of all ones whose order depends on the context where it is used. A generalised tensor factorisation problem is specified by an observed tensor  $X$  (with possibly missing entries, to be treated later) and a collection of latent tensors to be estimated,  $Z_1: \dots: Z_N = \{\mathcal{Z}^n\}$  for  $n = 1 \dots N$ , and by an exponential family of form (2). The index set of  $X$  is denoted by  $V_0$  and  $S$  the index set of each  $Z^n$  by  $V^n$ . The set of all model indices is  $V = \bigcup_{n=1}^N V^n$ . We use  $\mathbf{v}$  (or  $\mathbf{v}_0$ ) to denote a particular configuration of the indices for  $Z^n$  (or  $X$ ) while  $\mathbf{v}^n$  denoting a configuration of the complement  $V \setminus V^n = V/V^n$ . The goal is to find the latent  $Z^n$  that maximize the likelihood  $\mathcal{L}$  is given via  $p(X|Z_1: \dots: Z_N)$  where  $h(\mathbf{x}) = X(\mathbf{x})Y(\mathbf{x}) = g(X(\mathbf{v}_0 Z^n(\mathbf{v}^n))) = g(X(\mathbf{v}_0 Z^n(\mathbf{v}^n)))$

To clarify our notation with an example, we express the CP (Parafac) model, defined as  $X(i, j, k) = \sum_r Z(i, r)Z(j, r)Z(k, r)$ . In our notation, we take identity link  $g(X) = X$  and the index sets  $V_0 = \{i, j, k\}$ ,  $V^1 = \{r\}$ ,  $V^2 = \{i, r\}$ ,  $V^3 = \{j, r\}$  and  $V^4 = \{k, r\}$ . Our notation deliberately follows that of graphical models; the reader might find it useful to associate indices with discrete random variables and factors with probability tables [14]. Obviously, while a TF model does not represent a discrete probability measure, the algebraic structure is nevertheless analogous. To extend the discussion in Section 2.1 to the tensor case, we need the equivalent of the model matrix, when updating  $Z^n$ . This is obtained by summing over the product of all remaining factors  $X Y X X \dots = Z^n(\mathbf{v}^n) Z^n(\mathbf{v}^n) L^n(\mathbf{o}^n)$   $g(X(\mathbf{v}_0 Z^n(\mathbf{v}^n))) = \mathbf{v}_0 Z^n(\mathbf{v}^n) L^n(\mathbf{o}^n)$

$$\begin{aligned} L^n(\mathbf{o}^n) &= \\ X Y & \\ \mathbf{v}_0 Z^n(\mathbf{v}^n) Z^n(\mathbf{v}^n) &= \\ \mathbf{v}_0 Z^n(\mathbf{v}^n) & \\ \text{with } \mathbf{o}^n &= (\mathbf{v}_0 Z^n(\mathbf{v}^n)) \quad (\mathbf{v}_0 Z^n(\mathbf{v}^n)) \\ Z^n(\mathbf{v}^n) & \\ \mathbf{v}_0 Z^n(\mathbf{v}^n) Z^n(\mathbf{v}^n) &= \\ 3 \end{aligned}$$

$\nabla_{\mathbf{L}} \text{ wrt the latent tensor } \mathbf{Z}$  denoted as  $\text{One}$  related quantity to  $\mathbf{L}$  is the derivative of the tensor  $\mathbf{g}(\mathbf{X})$  and is defined as (following the convention [13, pp 196])  $\nabla_{\mathbf{L}} \mathbf{g}(\mathbf{X}) = \mathbf{I} - \mathbf{v}_0 \nabla_{\mathbf{v}} - \mathbf{L}$  with  $\mathbf{L} = \mathbf{R} - \mathbf{v}_0 \nabla_{\mathbf{v}} - \mathbf{v}_0 \nabla_{\mathbf{v}}$  — (10)  $\nabla_{\mathbf{L}} = \mathbf{Z}$ . The importance of  $\mathbf{L}$  is that, all the update rules can be formulated by a product and subsequent contraction of  $\mathbf{L}$  with another tensor  $\mathbf{Q}$  having exactly the same index set of the observed tensor  $\mathbf{X}$ . As a notational abstraction, it is useful to formulate the following function, Definition 1. The tensor valued function  $\nabla_{\mathbf{L}}(\mathbf{Q}) : \mathbf{R} - \mathbf{v}_0 - \mathbf{R} - \mathbf{v} - \mathbf{R} - \mathbf{v}$  is defined as  $\nabla_{\mathbf{L}}(\mathbf{Q}) =$

$$(11) \quad \mathbf{v}_0 \nabla_{\mathbf{v}} \mathbf{v}$$

with  $\nabla_{\mathbf{L}}(\mathbf{Q})$  being an object of the same order as  $\mathbf{Z}$  and  $\mathbf{o} = \mathbf{v}_0 \nabla_{\mathbf{v}} \mathbf{v}$  ( $\mathbf{v}_0 \nabla_{\mathbf{v}} \mathbf{v}$ ). Here, on the right side, the nonnegative integer  $\nabla$  denotes the element-wise power, not to be confused with an index. On the left, it should be interpreted as a parameter of the  $\nabla$  function. Arguably,  $\nabla$  function abstracts away all the tedious reshape and unfolding operations [5]. This abstraction has also an important practical facet: the computation of  $\nabla$  is algebraically (almost) equivalent to computation of marginal quantities on a factor graph, for which efficient message passing algorithms exist [14].  $\nabla_{\mathbf{L}} \mathbf{Q} = \mathbf{P}$  Example 1. TUCKER3 is defined as  $\mathbf{X} \mathbf{A}_i \mathbf{B}_j \mathbf{C}_k \mathbf{G}_{p,q,r}$  with  $\mathbf{V} = \{p, q, r\}$

$\{i, j, k, p, q, r\}$ ,  $\mathbf{V}_0 = \{i, j, k\}$ ,  $\mathbf{V}_A = \{i, p\}$ ,  $\mathbf{V}_B = \{j, q\}$ ,  $\mathbf{V}_C = \{k, r\}$ ,  $\mathbf{V}_G = \{p, q, r\}$ . Then for the first factor  $\mathbf{A}$ , the objects  $\mathbf{L}_A$  and  $\nabla_{\mathbf{L}_A}(\mathbf{Q})$  are computed as follows "  $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \mathbf{B} \mathbf{C} \mathbf{G} = ((\mathbf{C} \nabla_{\mathbf{L}_A} \mathbf{B}) \mathbf{G})$   $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \mathbf{B} \mathbf{C} \mathbf{G}$  (12)  $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \mathbf{B} \mathbf{C} \mathbf{G}$

$$(13) \quad \begin{aligned} & \nabla_{\mathbf{L}_A}(\mathbf{Q}) = \mathbf{B} \mathbf{C} \mathbf{G} \\ & \mathbf{X} \\ & \mathbf{p} \\ & \mathbf{Q}_{k,j} \mathbf{L}_A \mathbf{i} \\ & \mathbf{j}, \mathbf{k} \\ & \mathbf{k}, \mathbf{j} \\ & \mathbf{?} \\ & \mathbf{?} = \\ & \mathbf{Q} \mathbf{L} \mathbf{?} \mathbf{A} \\ & \mathbf{p} \mathbf{i} \end{aligned}$$

The index sets marginalised out for  $\mathbf{L}_A$  and  $\nabla_{\mathbf{L}_A}(\mathbf{Q})$  are  $\mathbf{V}_0 \nabla_{\mathbf{L}_A}(\mathbf{Q}) = \{p, q, r\}$   $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \{j, q, k, r\} = \{q, r\}$  and  $\mathbf{V}_0 \nabla_{\mathbf{L}_A}(\mathbf{Q}) = \{i, j, k\}$   $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \{j, q, k, r\} = \{j, k\}$ . Also we verify the order of the gradient  $\nabla_{\mathbf{L}_A}(\mathbf{Q})$  as  $\nabla_{\mathbf{L}_A}(\mathbf{Q}) = \mathbf{?}_{i,p} \mathbf{?}_{i,k,j}$  that conforms the matrix derivation convention [13, pp.196]. 2.3

#### Iterative Solution for GTF

As we have now established a one to one relationship between GLM and GTF objects such as the  $\nabla$  the model matrix  $\mathbf{L} = \mathbf{L}$  observation  $\mathbf{x} = \text{vec } \mathbf{X}$ , the mean (and the model estimate)  $\mathbf{x} = \text{vec } \mathbf{X}$ , and the parameter vector  $\mathbf{z} = \text{vec } \mathbf{Z}$ , we can write directly from (6) as  $\nabla_{\mathbf{L}}(\mathbf{Q}) =$

$$\nabla_{\mathbf{L}}(\mathbf{Q}) =$$

$Z = Z + \eta \frac{D(X - X)}{D(X - X)}$  (14) There are at least two ways that this update can further simplified. We may assume an identity link function, or alternatively we may choose a matching link and loss functions such that they  $\frac{D(X - X)}{D(X - X)} = 1$  that results to cancel each other smoothly [2]. In the sequel we consider identity link  $g(X) = g(X)$   $(X) = 1$ . This implies  $G$  to be identity, i.e.  $G = I$ . We define a tensor  $W$ , that plays the same role as  $w$  in (5), which becomes simply the precision (inverse variance function), i.e.  $W = 1/v(X)$  where for the Gaussian, Poisson, Exponential and Inverse Gaussian distributions we have simply  $\frac{D(X - X)}{D(X - X)} = p$  with  $p = \{0, 1, 2, 3\}$  [10, pp 30]. Then, the update (14) is reduced to  $W = X$  (1)

$Z = Z + \eta \frac{D(X - X)}{D(X - X)}$  (15) After this simplification we obtain two update rules for GTF for non-negative and real data.

The update (15) can be used to derive multiplicative update rules (MUR) popularised by [15] for the nonnegative matrix factorisation (NMF). MUR equations ensure the non-negative parameter updates as long as starting some non-negative initial values. 4

Theorem 1. The update equation (15) for nonnegative GTF is reduced to multiplicative form as  $Z = Z \frac{D(X - X)}{D(X - X)}$

$$\begin{aligned} & \frac{D(X - X)}{D(X - X)} (W \frac{D(X - X)}{D(X - X)}) \\ & \text{s.t. } Z \geq 0 \end{aligned} \quad (16)$$

(Proof sketch) Due to space limitation we leave the full details of the proof, but idea is that inverse of  $H = \frac{D(X - X)}{D(X - X)}$  is identified as step size and by use of the results of the Perron-Frobenius theorem [16, pp 125] we further bound it as

$\frac{D(X - X)}{D(X - X)} \leq \lambda_{\max}(H) \leq \lambda_{\max}(17) \leq v \leq \lambda_{\max}(\frac{D(X - X)}{D(X - X)})$  For the special case of the Tweedie family where the precision is a function of the mean as  $W = \frac{D(X - X)}{D(X - X)}$  for  $p = \{0, 1, 2, 3\}$  the update (15) is reduced to  $X = X \frac{D(X - X)}{D(X - X)}$

$$\frac{D(X - X)}{D(X - X)} (X \frac{D(X - X)}{D(X - X)}) \quad (18)$$

$\frac{D(X - X)}{D(X - X)} = Z \frac{D(X - X)}{D(X - X)}$ ,  $\frac{D(X - X)}{D(X - X)}$  is  $\frac{D(X - X)}{D(X - X)} = Z \frac{D(X - X)}{D(X - X)}$ . Then for the For example, to update  $Z$  for the NMF model  $X = Z \frac{D(X - X)}{D(X - X)}$  Gaussian ( $p = 0$ ) this reduces to NMF-EU as  $Z = Z \frac{D(X - X)}{D(X - X)}$ . For the Poisson ( $p = 1$ )  $\frac{D(X - X)}{D(X - X)}$

$$\frac{D(X - X)}{D(X - X)} \leq 1 \quad [15]. \text{ it reduces to NMF-KL as } Z = Z \frac{D(X - X)}{D(X - X)} \frac{D(X - X)}{D(X - X)}$$

By dropping the non-negativity requirement we obtain the following update equation: Theorem 2. The update equation for GTF with real data can be expressed as  $Z = Z + \eta \frac{D(X - X)}{D(X - X)}$

$$\begin{aligned} & \eta \frac{D(X - X)}{D(X - X)} (W \frac{D(X - X)}{D(X - X)}) \\ & \text{with } \frac{D(X - X)}{D(X - X)} = -v \frac{D(X - X)}{D(X - X)} \end{aligned} \quad (19)$$

(Proof sketch) Again skipping the full details, as part of the proof we set  $Z = 1$  in (17) specifically, 2 and replacing matrix multiplication of  $\frac{D(X - X)}{D(X - X)}$  by  $\frac{D(X - X)}{D(X - X)}$  completes the proof. Here the multiplier  $\frac{D(X - X)}{D(X - X)}$  is the cardinality arising from the fact that only  $\frac{D(X - X)}{D(X - X)}$  elements are non-zero in a row of  $\frac{D(X - X)}{D(X - X)}$ . Note the example for  $\frac{D(X - X)}{D(X - X)}$  that if  $V = \{p, q\}$  then  $\frac{D(X - X)}{D(X - X)} = -p - q$  which

is number of all distinct configurations for the index set  $\{p, q\}$ . Missing data can be handled easily by dropping the missing data terms from the likelihood [17]. The net effect of this is the addition of an indicator variable  $m_i$  to the gradient  $\partial L / \partial z = \sum_i (x_i - \hat{x}_i) m_i w_i g(x_i)$  with  $m_i = 1$  if  $x_i$  is observed otherwise  $m_i = 0$ . Hence we simply define a mask tensor  $M$  having the same order as the observation  $X$ , where the element  $M(v_0)$  is 1 if  $X(v_0)$  is observed and zero otherwise. In the update equations, we merely replace  $W$  with  $W \odot M$ .

3

### Coupled Tensor Factorization

Here we address the problem when multiple observed tensors  $X^{(i)}$  for  $i = 1, \dots, I$  are factorised simultaneously. Each observed tensor  $X^{(i)}$  now has a corresponding index set  $V_0^{(i)}$  and a particular configuration will be denoted by  $v_0^{(i)}, u^{(i)}$ . Next, we define a coupling matrix  $R$  where

$XY^{(i)}, 1 \leq i \leq I$  and  $Z^{(i)}$  connected  $u^{(i)}, v^{(i)} = R^{(i)}, X^{(i)} Z^{(i)}(v^{(i)}) R^{(i)}$  otherwise  $u^{(i)}, v^{(i)}$

?

For the coupled factorisation, we get the following expression as the derivative of the log likelihood

$X^{(i)} X^{(j)} \odot L^{(i,j)}(u^{(i)}, v^{(j)}) W^{(i,j)}(u^{(i)}, v^{(j)}) = (21) R^{(i,j)} X^{(i)}(u^{(i)}) X^{(j)} Z^{(j)}(v^{(j)}) Z^{(j)}(v^{(j)}) u^{(i)}, v^{(j)}$

?

$\gamma^{(i,j)}(u^{(i)}, v^{(j)})$  are the precisions. Then proceeding as in section 2.3 (i.e. getting where  $W^{(i,j)} = W$  (X Hessian and finding Fisher Information) we arrive at the update rule in vector form as  $X^{(i)} \odot X^{(j)}$

$\gamma^{(i,j)} Z^{(j)} +$

$Z^{(i)} (22) R^{(i,j)} \gamma^{(i,j)} R^{(i,j)} \gamma^{(i,j)} D^{(i,j)} \gamma^{(i,j)} D^{(i,j)} X^{(i)} \odot X^{(j)}$

?

5

$Z_1$

...

$Z_I$

...  $Z_{I-1}$

$X_1$

...

$X_I$

...  $X_{I-1}$

A

B

C

D

$X_1$

$X_2$

$X_3$

E

Figure 1: (Left) Coupled factorisation structure where the arrow indicates the existence of the influence of latent tensor  $Z$  onto the observed tensor  $X$ . (Right) The CP/MF/MF coupled factorisation problem in 1.  $??$  )/ $?Z$ . The update equations for the coupled case are quite intuitive; we where  $??,? = g(X$  calculate the  $??,?$  functions defined as  $Y \text{ h } X \text{ i } ?,? \text{ } Q(u? \text{ ) } ???,? (Q) = (23)$   $Z?? (v?? \text{ ) } R u? \text{ } ?? \text{ } v?$

$$\begin{aligned}
& \text{for each submodel and add the results: Lemma 1. Update for non-negative} \\
& \text{CTF } R^{?}, ?^{?}, ?^{?} (W^{?} ? X^{?}) \\
& \quad ?^{?} ?^{?} ?^{?} W^{?} ? X^{?} ? R \\
& \quad P \\
& \quad Z^{?} ? Z^{?} ? P^{?} \\
& (24)
\end{aligned}$$

In the special case of a Tweedie family, i.e. for the distributions whose precision as  $W = X$ , the
  $P \rightarrow P$  update is  $Z \rightarrow Z + \frac{1}{2} \frac{X}{X+1}$ .

Lemma 2. General update for CTF  $Z \rightarrow Z + \frac{1}{2} \frac{X}{X+1}$

we plug  $W = X$

4  
Experiments  
Here we want to solve the CTF problem introduced (1), which is a coupled CP/MF/MF problem  $X X X ? i,j,k = ? j,p = ? j,q = X A_{i,r} B_{j,r} C_{k,r} X B_{j,r} D_{p,r} X B_{j,r} E_{q,r}$  (26) 1 2 3 r  
r  
r

We want to use the general update equation (25). This requires derivation of  $\partial \mathcal{L} / \partial \theta_i$  for  $i = 1$  (CP) and  $i = 2$  (MF) but not for  $i = 3$  since that  $\partial \mathcal{L} / \partial \theta_3$  has the same shape as  $\partial \mathcal{L} / \partial \theta_2$ . Here we show the computation for B, i.e. for Z2, which is the common factor “#”



$$\begin{aligned} & \# \\ & = QD? \\ (29) \end{aligned}$$

with  $Q(n)$  being mode- $n$  unfolding operation that turns a tensor into matrix form [5]. In addition, for  $\ell = 1$  the required scalar value  $\ell B/0$  is  $-r-$  here since  $VB \ell V\ell 0 = \{j, r\} \ell \{r\} = \{r\}$  noting that value  $\ell B/0$  is the same for  $\ell = 2, 3$ . The simulated data size for observables is  $-i- = -j- = -k- = -p- = -q- = 30$  while the latent dimension is  $-r- = 5$ . The number of iterations is 1000 with the Euclidean cost while the experiment produced similar results for KL cost as shown in Figure 2. A

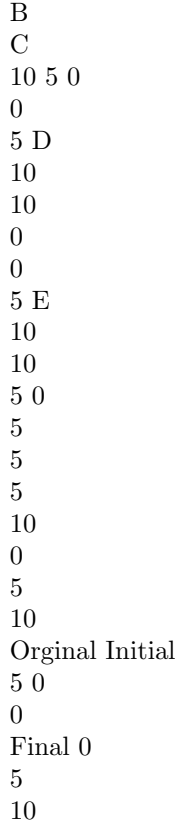


Figure 2: The figure compares the original, the initial (start up) and the final (estimate) factors for  $Z = A, B, C, D, E$ . Only the first column, i.e.  $Z(1 : 10, 1)$  is plotted. Note that CP factorisation is unique up to permutation and scaling [5] while MF factorisation is not unique, but when coupled with CP it recovers the original data as shown in the figure. For visualisation, to find the correct permutation, for each of  $Z$  the matching permutation between the original and estimate are found by solving an orthogonal Procrustes problem [18, pp 601].

#### 4.1 Audio Experiments

In this section, we illustrate a real data application of our approach, where we reconstruct missing parts of an audio spectrogram  $X(f, t)$ , that represents the STFT coefficient magnitude at frequency bin  $f$  and time frame  $t$  of a piano piece, see top left panel of Fig.3. This is a difficult matrix completion problem: as entire time frames (columns of  $X$ ) are missing, low rank reconstruction techniques are likely to be ineffective. Yet such missing data patterns arise often in practice, e.g., when packets are dropped during digital communication. We will develop here a novel approach, expressed as a coupled TF model. In particular, the reconstruction will be aided by an approximate musical score, not necessarily belonging to the played piece, and spectra of isolated piano sounds. Pioneering work of [19] has demonstrated that, when a audio spectrogram of music is decomposed  $X(f, t) = P D(f, i) E(i, t)$ , the computed factors  $D$  and  $E$  tend to be using NMF as  $X_1(f, t) = X(f, i)$  semantically meaningful and correlate well with the intuitive notion of spectral templates (harmonic profiles of musical notes) and a musical score (reminiscent of a piano roll representation such as a MIDI file). However, as time frames are modeled conditionally independently, it is impossible to reconstruct audio with this model when entire time frames are missing. In order to restore the missing parts in the audio, we form a model that can incorporate musical information of chords structures and how they evolve in time. In order to achieve this, we hierarchically decompose the excitation matrix  $E$  as a convolution of some basis matrices and their weights:  $E(i, t) = \sum_k B(i, k) C(k, t)$ . Here the basis tensor  $B$  encapsulates both vertical and temporal information of the notes that are likely to be used in a musical piece; the musical piece to be reconstructed will share  $B$ , possibly played at different times or tempi as modelled by  $G$ . After replacing  $E$  with the decomposed version, we get the following model (eq 30):  $X(f, t) = \sum_i D(f, i) B(i, k) C(k, d) Z(d, t)$  Test file (30)  $i, k, d$

$$X(f, t) = \sum_i D(f, i) B(i, k) C(k, d) Z(d, t)$$

$X$

$$B(i, k) G(k, m) Y(m, n)$$

MIDI file

(31)

Merged training files

(32)

$f, k, m$

$$X(f, p) = \sum_i D(f, i) F(i, p) T(i, p)$$

$X$

$$D(f, i) F(i, p) T(i, p)$$

$i$

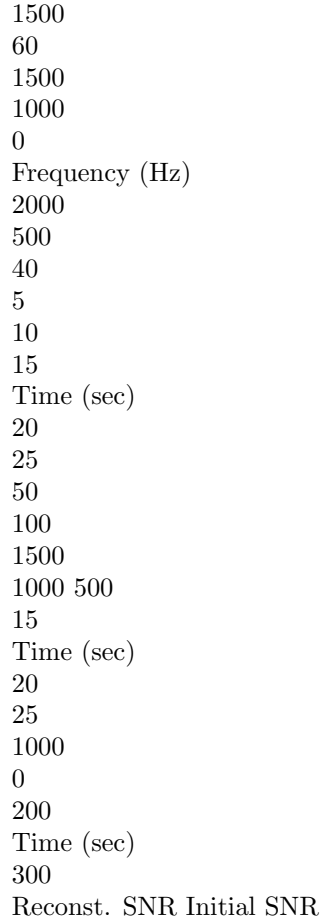
7

Here we have introduced new dummy indices  $d$  and  $m$ , and new (fixed) factors  $Z(d, t) = \delta(d - t)$  and  $Y(m, n) = \delta(m - n)$  to express this model in our framework. In eq 32, while forming  $X_3$  we concatenate isolated recordings corresponding to different notes. Besides,  $T$  is a  $0/1$  matrix, where  $T(i, p) = 1(0)$  if the note  $i$  is played (not played) during the time frame  $p$  and

F models the time varying amplitudes of the training data. R matrix for this model is defined as

$$\begin{aligned}
 &R = \\
 &\begin{pmatrix}
 1 & 1 & 0 & 1 & 1 & 0 \\
 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
 0 & 0 & 1 & & & & & & \\
 ? & 1 = P & D1 & B & 1 & C & 1 & Z & 1 & G0 & Y & 0 & F & 0 & T & 0 & X & ? & 2 = P & D0 & B & 1 & C & 0 & Z & 0 & G1 & Y & 1 \\
 F & 0 & T & 0 & \text{with } X & ? & 3 = P & D1 & B & 0 & C & 0 & Z & 0 & G0 & Y & 0 & F & 1 & T & 1 & X \\
 \# \\
 (33)
 \end{pmatrix}
 \end{aligned}$$

? ?1 ) on a 30 second piano Figure 3 illustrates the performance the model, using KL cost (W = X recording where the 70% of the data is missing; we get about 5dB SNR improvement, gracefully degrading from 10% to 80% missing data: the results are encouraging as quite long portions of audio are missing, see bottom right panel of Fig.3. X2 (Transcription Data)



15  
 10  
 5  
 500  
 10  
 100  
 Performance  
 SNR (dB)  
 1500  
 Frequency (Hz)  
 2000  
 5  
 0  
 150  
 Time (sec)  
 Ground Truth  
 2000  
 0  
 1000 500  
 20  
 X1hat (Restored)  
 Frequency (Hz)  
 X3 (Isolated Recordings)  
 80  
 Notes  
 Frequency (Hz)  
 X1 2000  
 5  
 10  
 15  
 Time (sec)  
 20  
 25  
 0  
 20  
 40  
 60  
 Missing Data Percentage (%)  
 80

Figure 3: Top row, left to right: Observed matrices X1 : spectrum of the piano performance, darker colors imply higher magnitude (missing data (70%) are shown white), X2 , a piano roll obtained from a musical score of the piece, X3 , spectra of 88 isolated notes from a piano. Bottom Row: Reconstructed X1 , the ground truth, and the SNR results with increasing missing data. Here, initial SNR is computed by substituting 0 as missing values.

## Discussion

This paper establishes a link between GLMs and TFs and provides a general solution for the computation of arbitrary coupled TFs, using message passing primitives. The current treatment focused on ML estimation; as immediate future work, the probabilistic interpretation is to be extended to a full Bayesian inference with appropriate priors and inference methods. A powerful aspect, which we have not been able to summarize here is assigning different cost functions, i.e. distributions, to different observation tensors in a coupled factorization model. This requires only minor modifications to the update equations. We believe that, as a whole, the GCTF framework covers a broad range of models that can be useful in many different application areas beyond audio processing, such as network analysis, bioinformatics or collaborative filtering. ? ?ITAK grant number 110E292, Bayesian Acknowledgements: This work is funded by the TUB matrix and tensor factorisations (BAYTEN) and Bo?gaziç?i University research fund BAP5723. Umut ? ?ITAK. We also would like to thank to S?ims?ekli is also supported by a Ph.D. scholarship from TUB Evrim Acar for the fruitful discussions. 8

## 2 References

- [1] A. T. Cemgil, Bayesian inference for nonnegative matrix factorisation models, *Computational Intelligence and Neuroscience* 2009 (2009) 1?17.
- [2] A. P. Singh, G. J. Gordon, A unified view of matrix factorization models, in: *ECML PKDD'08, Part II*, no. 5212, Springer, 2008, pp. 358?373.
- [3] E. Acar, T. G. Kolda, D. M. Dunlavy, All-at-once optimization for coupled matrix and tensor factorizations, *CoRR* abs/1105.3422. arXiv:1105.3422.
- [4] Q. Xu, E. W. Xiang, Q. Yang, Protein-protein interaction prediction via collective matrix factorization, in: *In Proc. of the IEEE International Conference on BIBM*, 2010, pp. 62?67.
- [5] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, *SIAM Review* 51 (3) (2009) 455?500.
- [6] Y. K. Y?lmaz, A. T. Cemgil, Probabilistic latent tensor factorization, in: *Proceedings of the 9th international conference on Latent variable analysis and signal separation, LVA/ICA'10*, Springer-Verlag, 2010, pp. 346?353.
- [7] C. Fevotte, A. T. Cemgil, Nonnegative matrix factorisations as probabilistic inference in composite models, in: *Proc. 17th EUSIPCO*, 2009.
- [8] Y. K. Y?lmaz, A. T. Cemgil, Algorithms for probabilistic latent tensor factorization, *Signal Processing* (2011), doi:10.1016/j.sigpro.2011.09.033.
- [9] C. E. McCulloch, S. R. Searle, *Generalized, Linear, and Mixed Models*, Wiley, 2001.
- [10] C. E. McCulloch, J. A. Nelder, *Generalized Linear Models*, 2nd Edition, Chapman and Hall, 1989.
- [11] R. Kaas, Compound poisson distributions and glm?s, tweedie?s distribution, Tech. rep., Lecture, Royal Flemish Academy of Belgium for Science and the Arts, (2005).
- [12] A. Cichocki, R. Zdunek, A. H. Phan, S. Amari, *Non-negative Matrix and Tensor Factorization*, Wiley, 2009.
- [13] J. R. Magnus, H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 3rd Edition, Wiley, 2007.
- [14] M. Wainwright, M. I. Jordan,

Graphical models, exponential families, and variational inference, *Foundations and Trends in Machine Learning* 1 (2008) 1?305. [15] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, in: *NIPS*, Vol. 13, 2001, pp. 556?562. [16] M. Marcus, H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Dover, 1992. [17] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: *Advances in Neural Information Processing Systems*, Vol. 20, 2008. [18] G. H. Golub, C. F. V. Loan, *Matrix computations*, 3rd Edition, Johns Hopkins UP, 1996. [19] P. Smaragdis, J. C. Brown, Non-negative matrix factorization for polyphonic music transcription, in: *WASPAA*, 2003, pp. 177?180.