

Liver CT Annotation via Generalized Coupled Tensor Factorization

Beyza Ermis and A. Taylan Cemgil

Boğaziçi University, 34342, Bebek, İstanbul, Turkey,
beyza.ermis@boun.edu.tr, taylan.cemgil@boun.edu.tr

Abstract. This study deals with the missing answers prediction problem. We address this problem using coupled analysis of *ImageCLEF2014* dataset by representing it as a heterogeneous data, i.e., dataset in the form of matrices. We propose to use an approach based on probabilistic interpretation of tensor factorization models, i.e., Generalized Coupled Tensor Factorization, which can simultaneously fit a large class of matrix/tensor models to higher-order matrices/tensors with common latent factors using different loss functions. Numerical experiments demonstrate that joint analysis of data from multiple sources via coupled factorization gives high prediction performance.

Keywords: matrix factorization, coupled analysis, missing value prediction

1 Introduction

In this study, the task is to predict user expressed (UsE) features by using the computer generated (CoG) features. UsE features are some predefined questions related to liver, vessels, lesions and CoG features are the features which are derived from image itself. Our aim is to fill in a pre-prepared form and provide computer aided automatic annotation of liver CT volumes by using image features [1, 2].

We consider, for answer prediction, both KL-divergence and Euclidean distance-based cost functions as well as coupled matrix factorization models [3-5] using GCTF [6] framework.

2 Methodology

In this study, we use Generalized Coupled Tensor factorization framework[6] for coupled factorization of several tensors and matrices to fill in the missing entries in observed data. A generalized tensor factorization problem is specified by an observed tensor X with possibly missing entries and a collection of latent tensors to be estimated, $Z_{1:|\alpha|} = \{Z_\alpha\}$ for $\alpha = 1 \dots |\alpha|$.

GCTF framework is a generalization of the Probabilistic Latent Tensor factorization (PLTF) [7] to coupled factorization. The Probabilistic Latent Tensor

Factorization framework (PLTF) [7] is appeared as an extension of the matrix factorization model [3] and enables one to incorporate domain specific information to any arbitrary factorization model and provides the update rules for multiplicative expectation-maximization (EM) algorithms. In this framework, the goal is to compute an approximate factorization of a given higher-order tensor, i.e., a multiway array, X in terms of a product of individual factors Z_α as:

$$X(v_0) \approx \hat{X}(v_0) = \sum_{\bar{v}_0} \prod_{\alpha} Z_{\alpha}(v_{\alpha}), \quad (1)$$

where some of the factors are possibly fixed. Here, we define v as the set of all indices in a model, v_0 as the set of visible indices, v_{α} as the set of indices in Z_{α} , and $\bar{v}_{\alpha} = v - v_{\alpha}$ as the set of all indices not in Z_{α} and $\alpha = 1, \dots, K$ as the factor index. Since the product $\prod_{\alpha} Z_{\alpha}(v_{\alpha})$ is collapsed over a set of indices, the factorization is latent. The approximation problem is cast as an optimization problem where we minimize the divergence $d(X, \hat{X})$, where d is a divergence between the observed data X and model prediction $\hat{X}$. In applications, d is typically taken as Euclidean (EUC), Kullback-Leibler (KL) or Itakura-Saito (IS) [7].

Here, we have two-dimensional datasets so we use nonnegative variant of the matrix factorization model. The matrix factorization model can be defined in the PLTF notation as follows. Given a matrix X , its factorization model is defined as:

$$X(i, j) \approx \hat{X}(i, j) = \sum_r Z_1(i, r) Z_2(j, r) \quad (2)$$

where the index sets $V = \{i, j, k, r\}$, $V_0 = \{i, j\}$, $V_1 = \{i, r\}$ and $V_2 = \{j, r\}$.

The update equation for non-negative generalized matrix/tensor factorization is expressed as:

$$Z_{\alpha} \leftarrow Z_{\alpha} \circ \frac{\Delta_{\alpha}(M \circ \hat{X}^{-p} \circ X)}{\Delta_{\alpha}(M \circ \hat{X}^{1-p})} \quad s.t. \quad Z_{\alpha}(v_{\alpha}) > 0. \quad (3)$$

where $\circ$ is the Hadamard product (element-wise product), M is a 0 – 1 mask array with $M(v_0) = 1$ ($M(v_0) = 0$) if $X(v_0)$ is observed (missing). Here p determines the cost function, i.e., $p = \{0, 1, 2\}$ correspond to the β -divergence [5] that unifies EUC, KL, and IS cost functions, respectively. In this iteration, we define the tensor valued function $\Delta_{\alpha}(A)$ as:

$$\Delta_{\alpha}(A) = \sum_{\bar{v}_{\alpha}} A(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}) \quad (4)$$

$\Delta_{\alpha}(A)$ is an object, the same size of Z_{α} , obtained simply by multiplying all factors other than the one being updated with an object of the order of the data. Hence the key observation is that the Δ_{α} function is just computing a

Table 1: Update rules for different p values

p	Cost Function	Multiplicative Update Rule
0	Euclidean	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu)}$
1	Kullback-Leibler	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-1} \circ X_\nu)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu)}$

tensor product and collapses this product over indices not appearing in Z_α , which is algebraically equivalent to computing a marginal sum.

This update rule can be used iteratively for all non-negative Z_α and converges to a local minimum provided we start from some non-negative initial values. For updating Z_α , we need to compute the Δ function twice for arguments $A = M_\nu \circ \hat{X}_\nu^{-p} \circ X_\nu$ and $A = M_\nu \circ \hat{X}_\nu^{1-p}$. It is easy to verify that update equations for the KL-NMF (non-negative matrix factorization) problem (for $p = 1$) are obtained as a special case of (3).

The Generalized Coupled Tensor Factorization (GCTF) [6] model takes the PLTF model one step further where, in this case, we have multiple observed tensors X_ν that are supposed to be factorized simultaneously:

$$X_\nu(v_{0,\nu}) \approx \hat{X}_\nu(v_{0,\nu}) = \sum_{\bar{v}_{0,\nu}} \prod_{\alpha} Z_\alpha(v_\alpha)^{R^{\nu,\alpha}} \quad (5)$$

$$R^{\nu,\alpha} = \begin{cases} 1 & \text{if } X_\nu \text{ and } Z_\alpha \text{ connected} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $\nu = 1, \dots, |\nu|$ and R is a *coupling matrix* that is defined as in (6). Note that, distinct from PLTF model, there are multiple visible index sets $(v_{0,\nu})$ in the GCTF model, each specifying the attributes of the observed tensor X_ν .

The inference, i.e., estimation of the shared latent factors Z_α , can be achieved via iterative optimization (see [6]). For non-negative data and factors, one can obtain the following compact fixed point equation where each Z_α is updated in an alternating fashion fixing the other factors $Z_{\alpha'}$ for $\alpha' \neq \alpha$

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-p} \circ X_\nu)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{1-p})}. \quad (7)$$

Here p , as in (3), determines the cost function, (see Table 1). In this iteration, the key quantity is the $\Delta_{\alpha,\nu}$ function that is defined as follows:

$$\Delta_{\alpha,\nu}(A) = \left[\sum_{v_{0,\nu} \cap \bar{v}_\alpha} A(v_{0,\nu}) \sum_{\bar{v}_{0,\nu} \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R^{\nu,\alpha'}} \right] \quad (8)$$

3 Experiments and Results

3.1 Experimental Setting

In this section, we address the missing answers (UsE features) prediction task on user expressed (UsE) features. UsE features are provided by a radiologist and represented by a 73×6 cell array. In our experiments, we used one column of that data which stands for *annotation's values*. For the experiments, we choose and separate the questions into two groups. In the first group, we used the question set:

$S1 = \{3, 5, 7, 9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28\}$ and in the second group we used the question set:

$S2 = \{29, 30, 31, 38, 39, 40, 41, 44, 48, 60, 66, 68, 71\}$. The questions in the first group have answers between $\{0 - 3\}$ and the questions in the second group have binary values $\{0 - 1\}$. *ImageCLEF2014* (liver CT annotation task) [2] has the content of 50 Matlab data files for training and 10 Matlab data files for test that contains CoG and UsE for each sample. We generate observation sample-questions matrices X_1 and X_2 by using these samples and the questions. In the end, we have the matrix X_1 of size 60×21 and X_2 of size 60×13 . In matrix X_1 , an entry $X_1(i, k)$ indicates the value of the answer k for sample i . Likewise, an entry $X_2(i, p)$ indicates the value of the answer p for sample i in matrix X_2 .

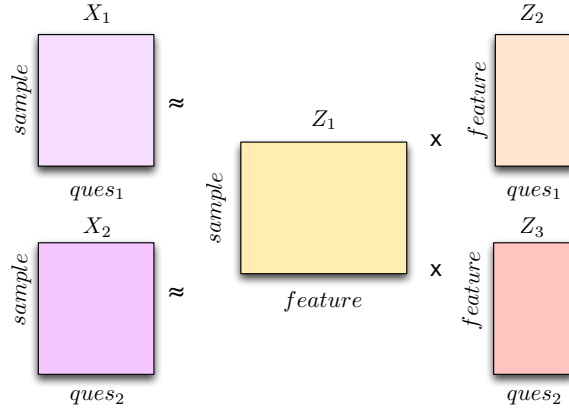


Fig. 1: General sketch of the proposed model. The blocks visualize the matrices that are defined in the model and the relation between them.

To predict the UsE features, we use Computer generated (CoG) features. These features are image-based features and they are generated by an interactive segmentation software. These features are stored in a 60×4 cell array.

Likewise UsE features, we used *feature's value* in our experiments. Here, we generate sample-features matrix Z_1 of size 60×447 . In matrix Z_1 , an entry $Z_1(i, j)$ indicates the value of the feature j for sample i .

Finally, we generate question-feature matrices Z_2 of size 447×21 and Z_3 of size 447×13 in order to complete the missing answers in X_1 and X_2 respectively.

The first 50 rows of X_1 and X_2 are observed and used for training while the rows from 51 to 60 are not observed and used for test. In order to fill in the missing entries in X_1 and X_2 , we applied the coupled approach to a CP-style tensor factorization model by analysing the tensor X_1 with X_2 . They have *samples* mode in common and this gives us the following model:

$$\begin{aligned}\hat{X}_1(i, k) &= Z_1(i, j)Z_2(j, k) \\ \hat{X}_2(i, p) &= Z_1(i, j)Z_3(j, p)\end{aligned}$$

(See Figure 1 for the visualization of the general structure of the model.)

After solving this model and obtain the values for latent factors, we construct the estimated matrices for X_1 and X_2 . Here, our goal is to predict ordinal values for answers on a scale of $\{1, \dots, R\}$. We can reduce this prediction task to a set of binary threshold problems, namely, predicting $r \geq 1, r \geq 2, \dots, r \geq R$. We try 1000 threshold values between 0–1, compute the accuracy of prediction for each value and choose the threshold value that gives the best accuracy performance. Threshold selection is not an easy task when we have different types of questions. For example, in the datasets there are several questions that their answers could have different values from all other questions. So, we have to find a unique value for that specific question. In this study, we ignore the questions with varied answers. That is why we have chosen these two sets of questions: to group the questions that have same type of answers is easier for this stage.

3.2 Results

Evaluation: Completeness and accuracy of the prediction are the two main evaluation criteria of this study. The predicted annotations are compared to the manual annotations of the test dataset. To measure the *completeness*, the number of predicted features are divided by total number of features and to measure the *accuracy*, the number of correct predicted features divided by total number of predicted features. Finally, the total score is given in terms of the ratio of correct answers in separate groups.

Table 2: Completion and accuracy results of previous method with EUC distance and KL divergence.

Cost Function	Completeness	Accuracy	Total Score
Kullback-Leibler (KL)	0.51	0.39	0.45

In the previous run we have used KL-divergence to estimate the factors and non-optimal threshold values to relate the real-valued predictions to the discrete predictions. We just choose a one threshold value manually, then predict all answers by using that value. At the end, we obtained the results given in Table 2.

Then, we used the thresholding method that we described in the Experimental Setting section. It determines the optimum value and we rerun the experiments by using this method. Here, we used both Euclidean distance and KL-divergence. Afterwards, we obtained the results with higher accuracy given in Table 3. We can clearly see that, our new method outperforms the previous one and the selection of the threshold significantly affects the prediction accuracy.

Table 3: Completion and accuracy results of our method with EUC distance and KL divergence.

Cost Function	Completeness	Accuracy	Total Score
Euclidean (EUC)	0.5151	0.8911	0.6775
Kullback-Leibler (KL)	0.5151	0.8882	0.6764

4 Conclusion

In this study, we have studied answer prediction problem using coupled analysis of *imageCLEF2014* data [2] represented as datasets in the form of matrices. The problem is formulated as simultaneous factorization of higher-order tensors/matrices extracting common latent factors from the shared modes. We have used Generalized Coupled Tensor Factorization framework, which enables us to develop coupled models for joint analysis of multiple data sets in a compact way using various models and cost functions. In our coupled analysis, we have studied both KL-divergence and Euclidean distance-based cost functions. In this study, we have used some questions and try to predict the values of their answers by using Euclidean distance and KL-divergence. We plan to extend our study to use different losses such as logistic loss or hinge loss and predict the answers of all questions in the dataset that have answers in a wide variety.

References

1. Marvasti, N., Kökciyan, N., Türkay, R., Yazıcı, A., Yolum, P., Üsküdarlı, S., Acar, B.: imageclef liver ct image annotation task 2014. In: CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. (2014)
2. Caputo, B., Müller, H., Martinez-Gomez, J., Villegas, M., Acar, B., Patricia, N., Marvasti, N., Üsküdarlı, S., Paredes, R., Cazorla, M., Garcia-Varea, I., Morell, V.: ImageCLEF 2014: Overview and analysis of the results. In: CLEF proceedings. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014)

3. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401** (1999) 788–791
4. Cemgil, A.T.: Bayesian inference in non-negative matrix factorisation models. *Computational Intelligence and Neuroscience* (Article ID 785152) (2009)
5. Cichoki, A., Zdunek, R., Phan, A., Amari, S.: *Nonnegative Matrix and Tensor Factorization*. Wiley (2009)
6. Yilmaz, Y.K., Cemgil, A.T., Simsekli, U.: Generalised coupled tensor factorisation. In: *NIPS*. (2011)
7. Yilmaz, Y.K., Cemgil, A.T.: Probabilistic latent tensor factorization. In: *LVA/ICA*. (2010) 346–353