TENSOR FACTORIZED DENSITY ESTIMATION

Simon Amtoft Pedersen, Marc Sun Bøg & Rasmus Tuxen
s173936, s173905 & s173910

Supervised by Morten Mørup

ABSTRACT

Estimating the densities of high dimensional datasets is a crucial role in unsupervised learning tasks. The purpose of this study is to investigate the use of tensor factorization methods to decompose the probabilistic mixtures to model such datasets. This has been done by implementing the Canonical Polyadic and Tensor Train decompositions as mixture models using the TensorFlow framework. The resulting models are able to compare against state-of-the-art density estimators and even surpass for a single dataset. This shows that tensor factorization methods for density estimation is feasible, and provide reasonable to good results on different datasets.

1. INTRODUCTION

Characterization of the density of data is a key challenge in unsupervised learning and is used in tasks such as simulating data, outlier detection and data imputation, which is at the core of probabilistic unsupervised learning and generative modelling. When performing density estimation a common obstacle is the curse of dimensionality, where the performance of methods worsens exponentially with high dimensional datasets.

A popular method for density estimation in machine learning is the use of probabilistic mixture models, where the curse of dimensionality is typically addressed through the use of suitable regularized parametric representations. In this paper, we investigate the the use of tensor decompositions [1] to combat the curse of dimensionality.

Tensors are multidimensional arrays and their decomposition can help with uncovering underlying hidden low-dimensional structure in the larger multidimensional tensor. Additionally, by using decompositions it is possible to create heterogeneous models, which uses different probabilistic distributions for the different dimensions.

This project will examine tensor factorized density estimators (TFDE) on mixture models in order to estimate the parameters for latent variable models by using tensor decompositions, which allows using heterogeneous mixtures instead of a single kind of mixture.

2. BUILDING MIXTURE MODEL DISTRIBUTIONS

In density estimation, mixture models is a probabilistic tool that mix multiple distributions together, such that it is possible to make statistical inferences about the underlying distribution of the data. For a general mixture model, we define *K* as the number of mixture components, *N* as the number of observations, and *M* as the number of dimensions in the data. In this section the TFDE methods are defined in terms of the homogeneous Gaussian Mixture model, however the same structure applies when regarding heterogeneous mixture models.

2.1. Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a probabilistic model which assumes that data points are generated from a mixture of K different Gaussian distributions. The GMM has a defined number of components, K, and for each component has a center $\boldsymbol{\mu}_k$ and full covariance matrix $\boldsymbol{\sigma}_k$ for the corresponding Gaussian distributions that it models, resulting in a complexity of $\mathcal{O}(KM^2)$. The probability of some data **x** is then calculated as (1), where $p(k)$ is the prior probability of component *k*, such that $\sum p(k) = 1$.

$$p(x_1, ..., x_M) = \sum_{k=1}^{K} p(k) \;\; \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \tag{1}$$

2.2. Canonical Polyadic Decomposition

The Canoical Polyadic (CP) decomposition is a so-called rank decomposition which aims to express a tensor as the sum of a finite number of rank-one tensors, $U_k(i_k, \alpha)$, called canonical factors [2].

$$A(i_1, i_2, ..., i_d) = \sum_{\alpha=1}^{r} U_1(i_1, \alpha) U_2(i_2, \alpha) ... U_d(i_d, \alpha) \tag{2}$$

In the same way, a probability density can be decomposed into canonical factors. When using Gaussians as the canonical factors, an expression equivalent to a GMM with diagonal covariance is achieved, and can be expressed as (3).

$$p(x_1, ..., x_M) = \sum_{k=1}^{K} p(k) \prod_{m}^{M} \mathcal{N}(\mu_{km}, \sigma_{km}^2) \quad (3)$$

The CP-decomposition of the mixture model thus has an achieved complexity of $\mathcal{O}(KM)$. From this, a heterogeneous mixture model would be achieved by for some *m* replacing the Gaussian with a different probabilistic model.

2.3. Tensor Train Decomposition

A drawback with the CP decomposition is that the computation of the upper bound of summation, *r*, is an NP-hard problem and that low-tensor-rank approximations are not guaranteed due to local minima [3]. The Tensor Train (TT) decomposition addresses this problem by seeking a block generalization of the rank-1 tensor [4]. Its computation is based on low-rank approximation of auxiliary unfolding matrices to construct the tensor (4).

$$A(i_1, ..., i_M) = \sum_{k_0, k_1, ..., k_M}^{K_0, K_1, ..., K_M} G_1 G_2 \dots G_M \quad (4)$$

Here, G_j represents an element, $C_j(k_{j-1}, i_j, k_j)$, from a 3D core tensor, for $j = 1, ..., M$, where the auxiliary indices k_j represent a link between auxiliary matrices.

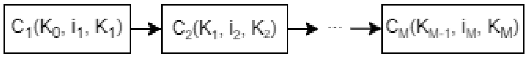

Fig. 1. A simple visualization of the Tensor Train network.

Like with the CP-decomposition, this concept can be applied to probability density estimation. Let the first core be $p(k_0)$, the probabilities of each cluster of a mixture model. Then the remaining product of cores, $\prod_{m=1}^{M} p(x_m, k_m | k_{m-1})$, consists of the underlying distributions for the dimensions of the final probability distribution, resulting in the distribution (5).

$$p(x_1, ..., x_M) = \sum_{k_0, ..., k_M}^{K_0, ..., K_M} p(k_0) \prod_{m=1}^{M} p(x_m, k_m | k_{m-1}) \quad (5)$$

3. IMPLEMENTATION & METHODOLOGY

From the definition of the TT mixture model decomposition (5), it is seen that the relative likelihood for a given point can be computed as

$$w_0 \; (W_1 \odot D_1)^T (W_2 \odot D_2)^T \dots (W_M \odot D_M)^T \; u \quad (6)$$

where *u* is a vector of ones, $\odot$ is the Hadamard product, w_0 is a $(1, K_0)$ column vector, W_m is a (K_m, K_{m-1}) weight matrix, and D_m is a (K_m, K_{m-1}) matrix of likelihood outputs from the underlying distributions.

Similarly, from (3), it is seen that the CP-decomposition can be trivially implemented as

$$(w \odot d_1 \odot d_2 \odot ... \odot d_M) \; u \quad (7)$$

where w is the cluster weights, d_m are corresponding likelihoods for each dimension, and u is a vector of ones.

These decomposition models are then implemented using the TensorFlow and TensorFlow Probability libraries[1], and fit like standard machine learning models using stochastic gradient descent on the negative log-likelihood.

One of the benefits of the TFDE models is their ability to tailor a density estimator to data. The two models loop over each dimension individually, which means that they can each be fit with different, suitable distribution. The ability to estimate densities of both continuous and discrete features can be trivially done, as each dimension of the CP- and TT-decompositions are configurable, which is referred to as heterogeneous modelling.

4. EXMPERIMENTAL RESULTS

4.1. Visualization of TFDE Models

To initially evaluate the different density models, and visualize how they fit different data distributions, the CP and TT models have been fit to different toy data sets which are generated as described in the FFJORD paper [5], which can be seen on figure 2.

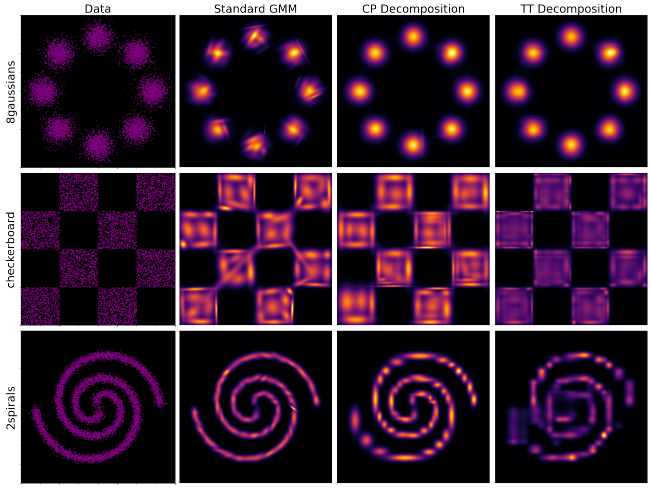

Fig. 2. Density plots for the standard GMM model, and CP- and TT-decompositions models, after being fit to three different toy data distributions.

[1] github.com/simonamtoft/tfde-tfp

From figure 2 it is seen that the GMM and the CP- and TT-decompositions fit the data pretty well. As expected, the GMM and CP densities are very similar, but for sharp density transitions like the `checkerboard` data, the TT model is better at separating density squares. However, as seen in the `2spirals` it models less rigidly structured densities worse.

A strength of the TFDE models is that they can be heterogeneous. A visualization of the heterogeneous TT and CP models against the three Gaussian TT models have been made on the checkerboard data, where a third dimension is added which defines a category for each square (figure 3). The models have all been trained having the same order of parameters, and thus the heterogeneous models have a clear advantage of being a single, collected model instead of training a different model for each category.

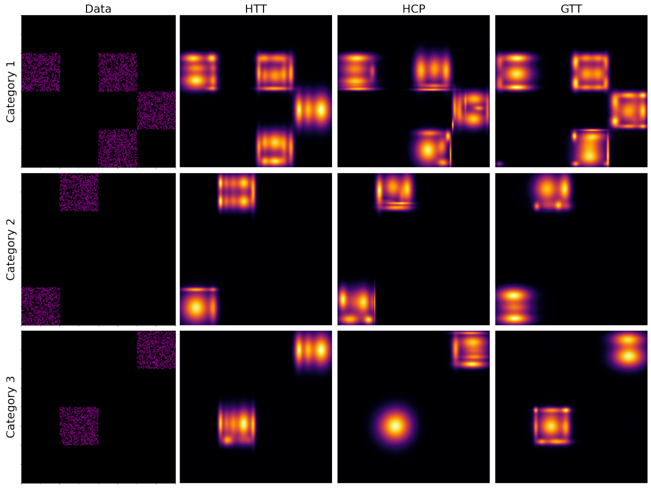

Fig. 3. A 3D plot of the checkerboard 2D toy data split into three categories, with each row representing a different category. The first column is the data. Second and third columns are heterogeneous TT and CP models. Fourth column is three different 2D TT models.

4.2. TFDE Model Results on High Dimensional Datasets

In this section, density estimation is performed on five tabular datasets and two image dataset from [6] and compared to one of the state-of-the-art methods described in the FFJORD paper [5]. The test error of both models is reported in table 1, where the hyper-parameters was selected using holdout cross-validation on a small subset of the data. To investigate the expressibility of the TFDE models, a test was performed on small subsets of one of the high-dimensional dataset. As seen in figure 4 the TT model is able to better utilize its trainable parameters in the high-dimensional space.

Dataset	TFDE (TT)	TFDE (CP)	FFJORD
POWER	-0.02	0.01	-0.46
GAS	-3.95	-5.44	-8.59
HEPMASS	22.38	23.60	14.92
MINIBOONE	33.29	41.43	10.43
BSDS300	-130.31	-127.47	-157.40
MNIST	0.06	2.57	1.05
CIFAR10[2]	N/A	N/A	3.40

Table 1. Negative log-likehood on test data for density estimation models; **lower is better**. In nats for tabular data and bits/dim. FFJORD results are from [5].

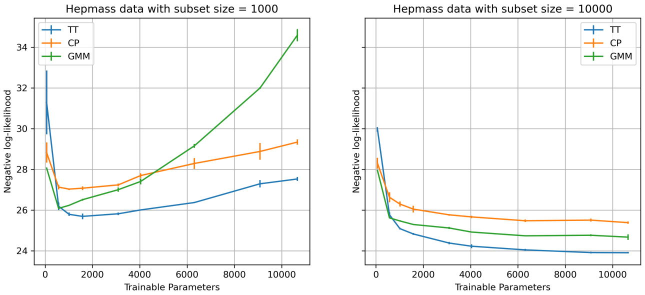

Fig. 4. Learning rate for Tensor train, Canonical Polyadic and Gaussian mixture model with full covariance trained on different subsets of the Hepmass dataset (23 dimensions, and $\approx 700,000$ samples) with the same number of free parameters. The error is measured on the full test set from the Hepmass dataset.

4.2.1. *Heterogeneous Data*

The heterogeneous TFDE models were trained on the Adult dataset, which has 15 dimensions, where several are categorical. The TFDE models were used to build a Bayesian Classifier on the dataset. In table 2 the different tensor decomposition methods are compared against a simple decision tree classifier (CART) as a proof of concept.

	TT	CP	CART
True Positive Rate	54.27 %	37.69 %	59.57 %
True Negative Rate	91.46 %	94.56 %	87.36 %
Accuracy	82.51 %	80.87 %	80.67 %

Table 2. Results from classification task on test set from Adult data for the heterogeneous TT and CP model and a simple classification tree. The prevalence of negative is 75.2 %.

Additionally, another advantage of the the trained TFDE models is that they can be used for imputation. In the appendix (figure 5) samples from the TFDE models were generated and it can be seen that both decompositions estimate most of the features fairly well.

[2] The CIFAR10 dataset was not evaluated due to an implementation specific issue. See discussion.

5. DISCUSSION

When comparing the performance of the different density estimators, the number of trainable parameters was selected to be about the same, to enable a fair comparison of the models. But while the number of trainable variables of the models are not exactly equivalent to the number of free parameters and thus does not fully reflect how much expressive power each model has, it is a metric that is easy to calculate for any type of model and allows for relatively fair comparisons, especially when also considering by-products such as memory and compute power required.

With that in mind, the TFDE models have been shown to be more versatile than the standard GMM, especially on higher-dimensional datasets where the TT model is able to better utilize the trainable parameters and thus be more expressible (figure 4). This is due to the fact that the TT model scales far better for increasing dimensionality, since the number of parameters increases linearly for the TT model but quadratically for the GMM model with full covariance, which allows better utilization of the parameters of the TT model. Additionally, the TT model can take advantage of dependencies between the neighbouring features in the data.

The performance of both of the TFDE models have been evaluated and compared to the state-of-the-art results presented in the FFJORD paper (see table 1). In comparison both the TT and CP implementations presented in this paper shows reasonable results for the presented datasets, but specifically for the MNIST dataset where the TT model was able to outperform FFJORD. Moreover, the TT model outperformed the CP model for all higher dimensional dataset, but for lower dimensions the two models are about equal in performance.

Furthermore, it is shown that heterogeneous modelling with the TT and CP models can provide substantial benefits on datasets by modelling with specified distributions for each dimension. Adding on to this, a proof of concept Bayesian classifier was trained on a dataset containing both continuous and discrete features for both heterogeneous TFDE models with a combination of Gaussian and categorical distributions. The resulting classifiers were compared against a simple classification tree, which shows that the decomposition models at the very least are capable of estimating heterogeneous densities well enough to yield the same accuracy as a classification tree.

A complication with the implemented TFDE models, and especially the TT model, is the increased computational time of the log-domain calculations, which is done to improve the numerical stability. The resulting implementation for matrix multiplication creates a large memory overhead that scales with K^3, which is caused by the broadcasting semantics of TensorFlow. This puts a limit on the size of the batches during training, thus making the computation time rather long. Additionally, a GPU TensorFlow specific issue with the implementation makes the memory overhead too large for very high-dimensional datasets, which is why it was impossible to produce any results for the `CIFAR10` dataset.

6. CONCLUSION

In this paper concepts from tensor factorization have been applied to probabilistic mixture modelling with the aim of estimating the density of data. These concepts are used to create two different Tensor Factorization Density Estimation models, which is based upon the Canonical Polyadic and Tensor Train decompositions. The resulting models are able to provide exact log-likelihood estimations along with generating new samples.

Contrary to other mixture models the TFDE models can utilize different distribution types depending on user specifications. This has enabled the TT and CP models to model heterogeneously which can provide substantial benefits on datasets with varying distributions over the different dimensions.

In comparison to a state-of-the-art density model, both the TT and CP implementations yield reasonable results for all of the evaluated datasets, where the performance of the TT model scaled better for increasing dimensionality compared to the CP model. Specifically for the MNIST dataset, the TT model outperformed the state-of-the-art density estimation. These results shows the promise of using tensor factorized mixture models especially when modelling high dimensional data with interdependent features.

7. REFERENCES

[1] Stephan Rabanser, Oleksandr Shchur, and Stephan Günnemann, "Introduction to tensor decompositions and their applications in machine learning," 2017.

[2] Felipe Bottega Diniz, "A fast implementation for the canonical polyadic decomposition," 2019.

[3] Johan Håstad, *Tensor rank is NP-complete*, vol. 11, pp. 451–460, Journal of Algorithms, 04 2006.

[4] I. V. Oseledets, "Tensor-train decomposition," *Society for Industrial and Applied Mathematics*, 2009.

[5] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud, "FFJORD: free-form continuous dynamics for scalable reversible generative models," *CoRR*, vol. abs/1810.01367, 2018.

[6] George Papamakarios, Theo Pavlakou, and Iain Murray, "Masked autoregressive flow for density estimation," 2018.

A. FIT ADULT DATA

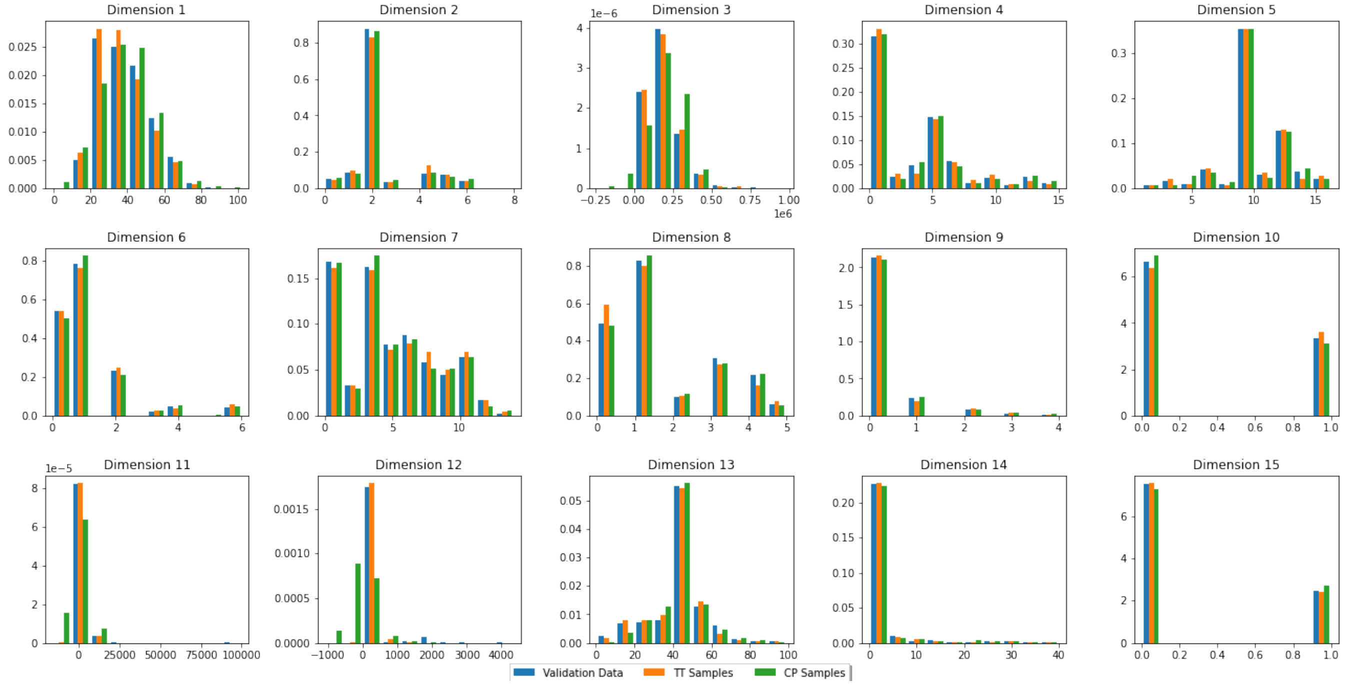

Fig. 5. Sampling of CP and TT heterogeneous TFDE models trained on the Adult dataset, which contains 14 features and a label.