



# Input-Aware Sparse Tensor Storage Format Selection for Optimizing MTTKRP

Hailong Yang<sup>1</sup>, Yi Liu<sup>1</sup>, and Zhongzhi Luan<sup>1</sup>, Beihang University

Lin Gan<sup>2</sup> and Guangwen Yang, Tsinghua University

Depei Qian<sup>1</sup>, Beihang University

*This installment of Computer's series highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Computers.*

**V**arious sparse tensor formats have been proposed to optimize the performance of tensor computations. For the first time, we leverage unsupervised machine learning methods

to automatically select the optimal sparse storage format for tensor computations. Our proposed framework can achieve high prediction accuracy and thus significant performance speedup in practical applications.

## SUMMARY

Tensors can represent high-dimensional data with more than two dimensions. Tensor decomposition is widely used to understand the relationship of data across multiple dimensions. Canonical polyadic decomposition (CPD) is a generalization of singular value decomposition and outputs matrix factors for

each mode of a tensor. The major performance bottleneck of CPD is matricized tensor times Khatri-Rao product (MTTKRP), which is the primary focus of optimizations in tensor composition.

Existing works optimize the performance of MTTKRP based on the computation patterns and operation dependency. Although the parallelization can significantly

Digital Object Identifier 10.1109/MC.2023.3279447  
Date of current version: 26 July 2023



improve the performance of MTKRP, it is constrained by the sparsity patterns and hardware characteristics. Therefore, different sparse tensor formats have been proposed to improve the computation performance with codesigned storage and algorithms adapted to the sparsity and hardware. However, due to the complex sparsity patterns and diverse hardware characteristics, the optimal tensor format varies significantly.

The format selection of sparse tensors can be analogized to the classification problem. For programmers, choosing the optimal format is a daunting task requiring tedious efforts. The convolutional neural network (CNN) has gained tremendous popularity in classification tasks due to its ability to capture the underlying features of input data. However, CNNs cannot be directly applied in tensor format selection, due to higher-dimensional data to deal with. The high-dimensional convolution can neither be used due to the unacceptable prediction overhead caused by the tensor irregularity.

Unlike supervised methods, unsupervised methods only require unlabeled training data, which can significantly reduce engineering efforts. Among them, the convolutional autoencoder (CAE) has gained attention in classification tasks due to its ability to effectively extract the pixel distribution of an image as a feature vector. However, the same challenges faced by CNNs also apply to CAEs. In addition, a holistic pipeline including the autoencoder and clustering algorithm needs to be designed.

In our article,<sup>1</sup> we propose an automatic tensor format selection framework, SpTFS, which can

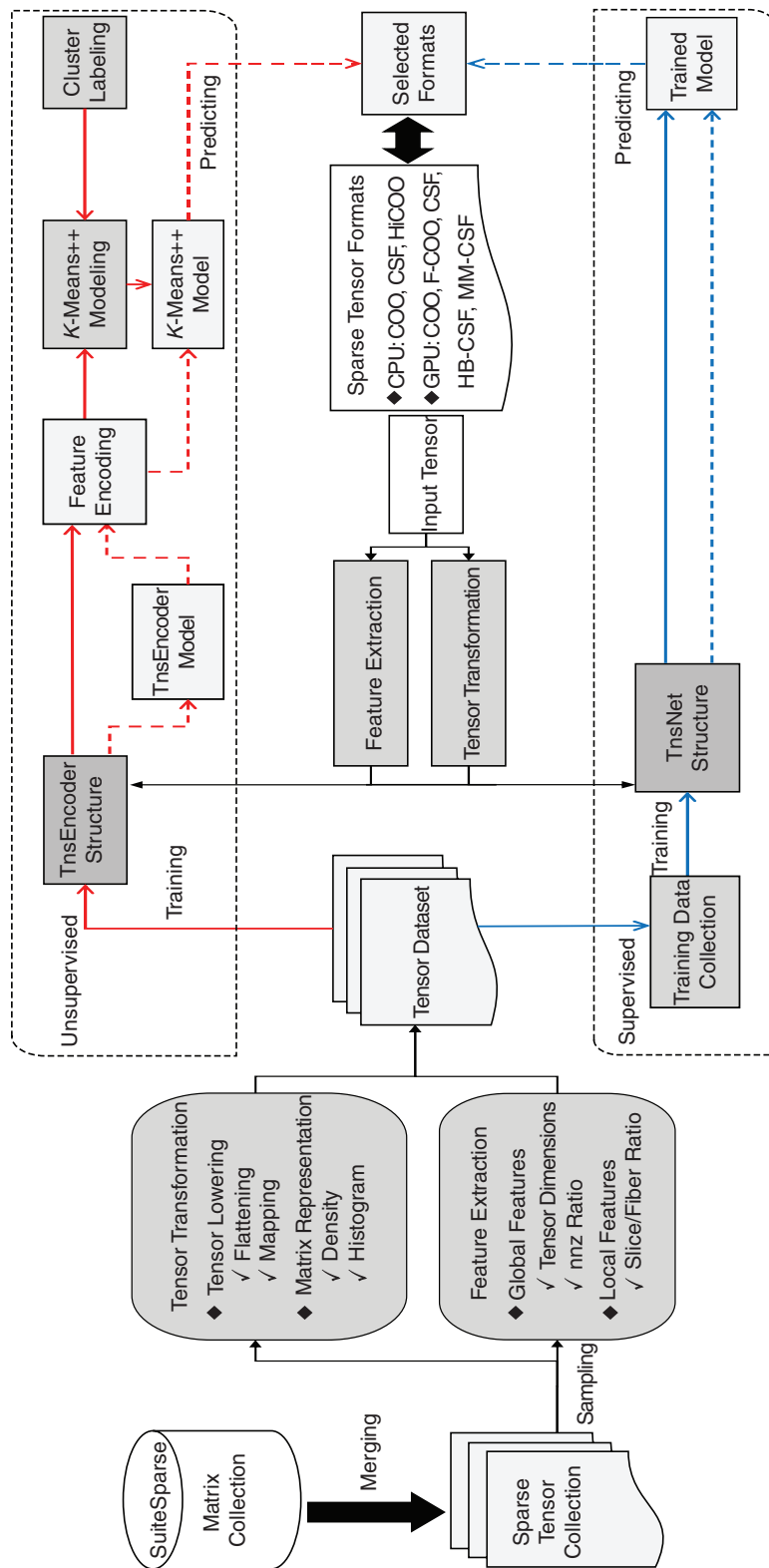


FIGURE 1. The design overview of SpTFS. nnz: number of non-zeros.

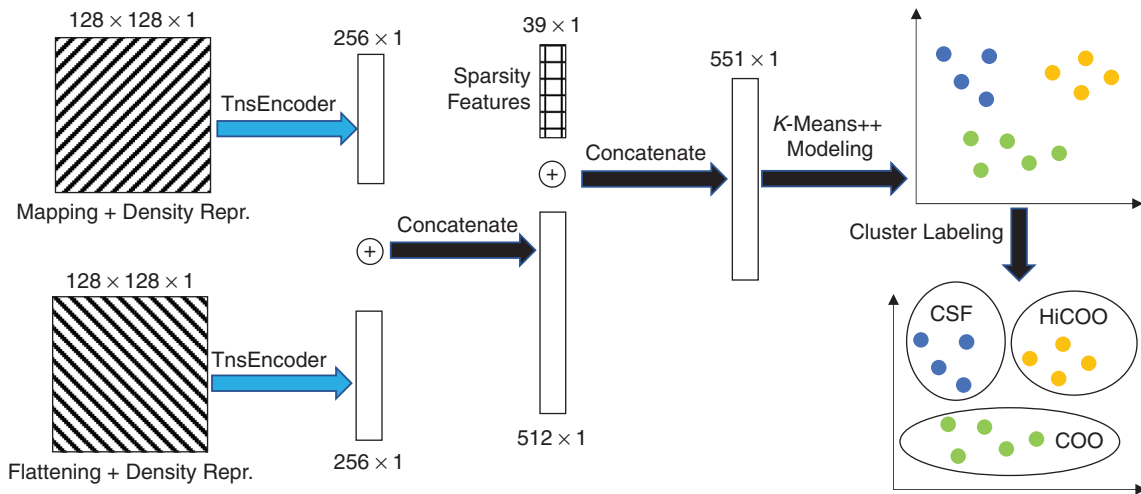


FIGURE 2. The design of TnsClustering for predicting the optimal tensor format. repr.: representation.

TABLE 1. Clustering accuracy comparison of unsupervised learning-based methods.

Architecture and dimension	MTTKRP mode	TnsClustering			FcClustering			PureClustering		
		Top 1 acc.	Top 2 acc.	H score	Top 1 acc.	Top 2 acc.	H score	Top 1 acc.	Top 2 acc.	H score
CPU and 3D	Mode 1	<b>0.75</b>	<b>0.95</b>	<b>0.58</b>	0.69	0.93	0.5	0.69	0.9	0.46
	Mode 2	<b>0.75</b>	<b>0.95</b>	<b>0.6</b>	0.73	0.95	0.57	0.7	0.91	0.49
	Mode 3	<b>0.77</b>	0.94	<b>0.5</b>	0.77	<b>0.95</b>	0.5	0.75	0.92	0.41
GPU and 3D	Mode 1	<b>0.72</b>	0.94	<b>0.41</b>	0.71	<b>0.95</b>	0.4	0.67	0.92	0.29
	Mode 2	<b>0.66</b>	<b>0.91</b>	<b>0.41</b>	0.62	0.89	0.36	0.6	0.86	0.3
	Mode 3	<b>0.64</b>	<b>0.89</b>	<b>0.43</b>	0.61	0.87	0.39	0.57	0.84	0.31
CPU and 4D	Mode 1	<b>0.64</b>	<b>0.9</b>	<b>0.38</b>	0.62	0.89	0.36	0.61	0.89	0.34
	Mode 2	<b>0.65</b>	<b>0.88</b>	<b>0.44</b>	0.62	0.87	0.4	0.61	0.87	0.39
	Mode 3	<b>0.63</b>	<b>0.89</b>	<b>0.41</b>	0.6	0.88	0.37	0.61	0.88	0.38
	Mode 4	<b>0.62</b>	<b>0.88</b>	0.4	0.6	0.88	0.38	0.62	0.88	<b>0.4</b>

The bold type indicates the best top 1, top 2, and h-score across the three clustering methods. acc.: accuracy.


predict the optimal tensor format for MTTKRP with redesigned CNN and CAE networks. As shown in Figure 1, the SpTFS sampling consists of two important components including tensor transformation and feature extraction. The tensor transformation component converts the sparse tensors into fixed-size matrices through tensor lowering and matrix representation. The feature

extraction component captures lost tensor features during tensor transformation, which are then fed into the fully connected layer.

For unsupervised learning, we propose TnsClustering, which consists of feature encoding, K-means++ modeling, and cluster labeling (depicted in Figure 2). During prediction, TnsClustering obtains the feature

vector of each input tensor through feature encoding. Then, the input tensor is assigned to the nearest cluster by the trained K-means++ model, with the format predicted the same as the cluster.

We evaluate SpTFS on both CPU and GPU platforms to prove its effectiveness in predicting the optimal tensor format. As reported in Table 1,

TnsClustering achieves average top 1/2 accuracies of 76%/95% and 67%/92% on CPU and GPU, respectively. In return, TnsClustering achieves 4.03× and 1.45× performance speedup of MTTKRP over the coordinate format on average. 

#### ACKNOWLEDGMENT

Hailong Yang is the corresponding author.

#### REFERENCE

1. Q. Sun et al., "Input-aware sparse tensor storage format selection for optimizing MTTKRP," *IEEE Trans. Comput.*, vol. 71, no. 8, pp. 1968–1981, Aug. 2022, doi: 10.1109/TC.2021.3113028.

**HAILONG YANG** is an associate professor in the School of Computer Science and Engineering, Beihang University, Beijing 100191, China. Contact him at [hailong.yang@buaa.edu.cn](mailto:hailong.yang@buaa.edu.cn).

**YI LIU** is a professor in the School of Computer Science and Engineering and the director of the Sino-German Joint Software Institute, Beihang University, Beijing 100191, China. Contact him at [yi.liu@buaa.edu.cn](mailto:yi.liu@buaa.edu.cn).

**ZHONGZHI LUAN** is an associate professor of computer science and engineering and the assistant director of the Sino-German Joint Software Institute Laboratory, Beihang University, Beijing 100191, China. Contact him at [07680@buaa.edu.cn](mailto:07680@buaa.edu.cn).

**LIN GAN** is an assistant researcher in the Department of Computer Science and Technology, Tsinghua University, China, and the assistant director of the National Supercomputing Center, Wuxi, Beijing 100083, China. Contact him at [lingan@tsinghua.edu.cn](mailto:lingan@tsinghua.edu.cn).

**GUANGWEN YANG** is a professor in the Department of Computer Science and Technology, Tsinghua University, China, and the director of the National Supercomputing Center, Wuxi, Beijing 100083, China. Contact him at [ygw@tsinghua.edu.cn](mailto:ygw@tsinghua.edu.cn).

**DEPEI QIAN** is a professor at the Department of Computer Science and Engineering, Beihang University, Beijing 100191, China. Contact him at [depei@buaa.edu.cn](mailto:depei@buaa.edu.cn).

## IEEE Computer Society Has You Covered!

**WORLD-CLASS CONFERENCES** — Over 189 globally recognized conferences.

**DIGITAL LIBRARY** — Over 893k articles covering world-class peer-reviewed content.

**CALLS FOR PAPERS** — Write and present your ground-breaking accomplishments.

**EDUCATION** — Strengthen your resume with the IEEE Computer Society Course Catalog.

**ADVANCE YOUR CAREER** — Search new positions in the IEEE Computer Society Career Center.

**NETWORK** — Make connections in local Region, Section, and Chapter activities.

Explore all of the member benefits at [www.computer.org](http://www.computer.org) today!

