SCIENCE CHINA
Information Sciences

• LETTER •

March 2024, Vol. 67, Iss. 3, 139404:1–139404:2
https://doi.org/10.1007/s11432-023-3869-3

Density peak clustering using tensor network

Xiao SHI[1,2] & Yun SHANG[1,3*]

[1]*Institute of Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3]*National Center for Mathematics and Interdisciplinary Sciences, Key Laboratory of Management, Decision and Information Systems, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China*

Received 3 March 2023/Revised 3 July 2023/Accepted 21 September 2023/Published online 20 February 2024

Tensor networks have been a powerful tool in simulating many-body physics and have recently gained recognition in the machine learning community due to their remarkable representation capabilities. However, using tensor networks to address the problem of clustering with an indeterminate number of clusters has yet to be explored.

Combining tensor networks and machine learning is an emerging field. Ever since Bény [1] and their contemporaries pioneered the fusion of tensor networks and machine learning in 2013, the landscape of this interdisciplinary domain has witnessed a burgeoning wealth of exceptional contributions. For example, Stoudenmire et al. [2] were inspired by the density matrix renormalization group (DMRG) algorithm and used tensor works for classification tasks. Han et al. [3] used tensor networks to generate models (see Appendix A for more related work).

In this study, we introduce a new approach to density-based clustering that utilizes matrix product state (MPS) to facilitate the identification of an unknown number of clusters in a data set. Our approach is rooted in the principles of density-peak clustering (DPC) [4], but incorporates the MPS representation to enable the capturing of data in a higher-dimensional Hilbert space. This leads to a more comprehensive representation of the data, enabling the identification of a hyperplane that can linearly separate the data, even if it is not linearly separable in the original space. Additionally, key properties such as density, core point, and border point are redefined using fidelity measures to enhance clustering performance. We evaluate this algorithm on a diverse range of data sets, including six synthetic data sets, four real-world data sets, and three computer vision data sets. Results demonstrate that this algorithm offers performance at the state-of-the-art level on some synthetic and real-world data sets, even when the number of clusters is unknown. It also performs comparably with state-of-the-art algorithms on the MNIST, USPS, and Fashion-MNIST image data sets. These findings highlight the potential of tensor networks for machine learning applications, and suggest that our approach may be a promising direction for further research.

Training MPS. In order to train an MPS with all the data, the first thing is to map all the data into quantum states. Suppose we have data set $S = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$. The i-th element $\boldsymbol{x}_i^k$ of the input vector $\boldsymbol{x}^k = (x_1^k, x_2^k, \ldots, x_m^k)$ of length m is mapped to a superposition of quantum states $|0\rangle$ and $|1\rangle$, which can be described as

$$|\psi_i^k\rangle = \cos\left(\frac{\pi}{2}x_i^k\right)|0\rangle + \sin\left(\frac{\pi}{2}x_i^k\right)|1\rangle. \tag{1}$$

Therefore, the input vector $\boldsymbol{x}^k$ can be written as the tensor product of $|\psi_i^k\rangle$

$$|\Psi(\boldsymbol{x}^k)\rangle = |\psi_1^k\rangle \otimes |\psi_2^k\rangle \otimes \cdots \otimes |\psi_m^k\rangle. \tag{2}$$

It is expressible in the form of a tensor network state, as

$$|\Psi_k^{\boldsymbol{\sigma}}\rangle = \sum_{\alpha_0,\alpha_1,\ldots,\alpha_m} X^{\sigma_0}_{k,\alpha_0,\alpha_1} X^{\sigma_1}_{k,\alpha_1,\alpha_2} \cdots X^{\sigma_{m-1}}_{k,\alpha_{m-1},\alpha_m} |\sigma\rangle, \tag{3}$$

where σ_i is its physical indices, α_i is its auxiliary indices with $\alpha_0 = \alpha_1 = \cdots = \alpha_n = 1$. Each X_i represents a $1 \times 2 \times 1$ third-order tensor whose elements are

$$X^1_{i,1,1} = \cos\left(\frac{\pi}{2}x_i^k\right), \quad X^2_{i,1,1} = \sin\left(\frac{\pi}{2}x_i^k\right). \tag{4}$$

Analogously, we randomly generate a quantum state $\Phi^{\boldsymbol{\tau}}$ of length m and a bond dimension equals to D in the form of MPS

$$|\Phi^{\boldsymbol{\tau}}\rangle = \sum_{\beta_0,\beta_1,\ldots,\beta_m} Y^{\tau_0}_{\beta_0,\beta_1} Y^{\tau_1}_{\beta_1,\beta_2} \cdots Y^{\tau_{m-1}}_{\beta_{m-1},\beta_m} |\tau\rangle, \tag{5}$$

where β is the auxiliary indices, which determines the upper limit of the entanglement entropy that this MPS state can accommodate, and $1 \leqslant \beta_i \leqslant D$. After completing the above steps, a variational matrix product states algorithm will be used to update the parameters in the MPS.

Gradient descent is used to update each tensor when the quantum state $\Phi^{\boldsymbol{\tau}}$ satisfies the normalization condition:

$$Y^{\tau_i} - \eta \frac{\partial f}{\partial Y^{\tau_i}} \to Y^{\tau_i}, \tag{6}$$

where η is the learning rate. Tensors will be updated from the first to the last, and then back, this process is called a sweep. The iteration process will stop when the sweep

* Corresponding author (email: shangyun@amss.ac.cn)

reaches the maximum number of iterations or the loss function converges. After this, the quantum state $\Phi^{\boldsymbol{\tau}}$ will give the joint distribution probability of pixels (see Appendix B.1 for details).

Generation of clusters. After training through the algorithm in the last section, a quantum state $\Phi^{\boldsymbol{\tau}}$ with bond dimension equal to D is obtained. Similar to the DPC algorithm, we define ρ and δ by fidelity measure instead of distance measure, and replace the Gaussian kernel function with the Sigmoid kernel function:

$$\rho_i = \tanh(f_i/(10 \times f_c)), \tag{7}$$

where $f_i = |\langle \Psi_i^{\boldsymbol{\sigma}} | \Phi^{\boldsymbol{\tau}} \rangle|$ represents the fidelity between quantum state $\Psi_i^{\boldsymbol{\sigma}}$ and $\Phi^{\boldsymbol{\tau}}$, f_c is a predefined cutoff distance. Similarly, computing the fidelity $f_{ij} = |\langle \Psi_i^{\boldsymbol{\sigma}} | \Psi_j^{\boldsymbol{\sigma}} \rangle|$ of the quantum states of the data points x_i and x_j, we get the definition of δ_i as follows:

$$\delta_i = \min_{j:\rho_j > \rho_i} (f_{ij}). \tag{8}$$

After this, we will give the definition of a local cluster center as follows:

Definition 1 (Local cluster center)**.** A data point x_i is defined as a local cluster center if it satisfies the conditions $\delta_i > f_c$ and $\rho_i > \overline{\rho}$, where $\overline{\rho}$ is the average of all point density.

After the local cluster centers have been identified, the remaining points can be assigned to their nearest higher-density neighbor to generate a set of local clusters. Therefore, L local clusters $(C^{(1)}, C^{(2)}, \ldots, C^{(L)})$ are obtained.

Similar to the density-based spatial clustering of applications with noise (DBSCAN) algorithm, for each local cluster $C^{(k)}$, the core point and border point need to be defined. Therefore, for the local cluster $C^{(k)}$, an MPS representation $\Phi_k^{\boldsymbol{\tau}}$ is trained. Then we define the core point and border point as follows.

Definition 2 (Definition of core point and border point)**.** Assuming that the point c_i in the local cluster C^k satisfies $f_i' > \overline{f}_k'$, where $f_i' = |\langle \Psi_i^{\boldsymbol{\sigma}} | \Phi_k^{\boldsymbol{\tau}} \rangle|$, $\overline{f}_k' = \frac{1}{n_k} \sum_{c_j \in C^{(k)}} |\langle \Psi_j^{\boldsymbol{\sigma}} | \Phi_k^{\boldsymbol{\tau}} \rangle|$, and n_k is the number of points in the local cluster C^k. Then the point c_i is called the core point of the local cluster C^k, otherwise c_i is the border point.

In the following, in order to determine the connectivity between local clusters, we will give the definitions of density directly-connectable and density connectable respectively (see Appendix B.2 for the definitions). Finally, all local clusters with density connectable are merged to get the final clustering result [5].

Results and analysis. In our quest to validate our findings, we standardized the bond dimension of MPS at 8 and conducted comprehensive experiments spanning synthetic, real-world, and computer vision datasets (see Appendix C for details). Our algorithm demonstrated remarkable performance on synthetic data, achieving a 100% clustering accuracy, with only slight degradation on imbalanced data. In real-world datasets, it excels, particularly on the Wine dataset, showcasing state-of-the-art results while also achieving the highest Fowlkes-Mallows index (FMI) on the Vehicle and Yeast datasets. Next, we extended our analysis to commonly employed image datasets, where we harnessed the power of autoencoder and uniform manifold approximation and projection (UMAP) methods for data preprocessing. Part of the results are shown in Table 1. Remarkably, our algorithm demonstrates state-of-the-art performance on the MNIST and USPS datasets without knowing the number of clusters. These results underscore the robustness and adaptability of our approach, signifying its potential as a valuable tool for data analysis and clustering across various domains.

Illustrating our approach using the Wine dataset as a prime example, we meticulously investigate the intricate interplay between entanglement entropy and accuracy. Additionally, we elucidated the rationale behind opting for the Sigmoid kernel function over the Gaussian kernel function, providing a clear and reasoned justification for this pivotal choice. Also, we proved our algorithm's time complexity is $O(kLD^3 + nLdD^2 + n^2L)$ (all the details can be found in Appendix D).

Table 1 Comparison of the accuracy of our algorithm and common algorithms on different data sets

Methods	MNIST (%)	USPS (%)	Fashion (%)
K-means	53.91	65.76	52.22
DBSCAN	–	16.7	10.0
DPC	–	39.9	34.4
MPS-8	97.8	97.1	59.99

Conclusion. We introduce a density-based clustering algorithm with tensor networks. In order to demonstrate its effectiveness, we apply it to various types of data sets, including synthetic data sets, real world data sets, and computer vision data sets. Results demonstrate that it is an efficient quantum-inspired unsupervised learning algorithm and can recognize clusters of arbitrary shape and size. It can also be seen that large quantum entanglement tends to provide better clustering results.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2023YFA1009403), National Natural Science Foundation Special Project of China (Grant No. 12341103), and National Natural Science Foundation of China (Grant No. 62372444).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Bény C. Deep learning and the renormalization group. 2013. ArXiv:1301.3124
2 Stoudenmire E, Schwab D J. Supervised learning with tensor networks. In: Proceedings of the 30th Conference on Neural Information Processing System, 2016
3 Han Z Y, Wang J, Fan H, et al. Unsupervised generative modeling using matrix product states. Phys Rev X, 2018, 8: 031012
4 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344: 1492–1496
5 Ren Y, Wang N, Li M, et al. Deep density-based image clustering. Knowl-Based Syst, 2020, 197: 105841