Fuzzy c-Means Clustering for Persistence Diagrams

Thomas Davies
University of Southampton
t.o.m.davies@soton.ac.uk

Jack Aspinall
University of Oxford
jack.aspinall@materials.ox.ac.uk

Bryan Wilder
Harvard University
bwilder@g.harvard.edu

Long Tran-Thanh
University of Southampton
L.Tran-Thanh@soton.ac.uk

Abstract

Persistence diagrams concisely represent the topology of a point cloud whilst having strong theoretical guarantees. Most current approaches to integrating topological information into machine learning implicitly map persistence diagrams to a Hilbert space, resulting in deformation of the underlying metric structure whilst also generally requiring prior knowledge about the true topology of the space. In this paper we give an algorithm for Fuzzy c-Means (FCM) clustering directly on the space of persistence diagrams, enabling unsupervised learning that automatically captures the topological structure of data, with no prior knowledge or additional processing of persistence diagrams. We prove the same convergence guarantees as traditional FCM clustering: every convergent subsequence of iterates tends to a local minimum or saddle point. We end by presenting experiments where the fuzzy nature of our topological clustering is capitalised on: lattice structure classification in materials science and pre-trained model selection in machine learning.

1 Introduction

Persistence diagrams, a concise representation of the topology of a point cloud with strong theoretical guarantees, have emerged as a new tool in the field of data analysis (Edelsbrunner and Harer, 2010). However, the non-Hilbertian nature of the space of persistence diagrams (Bubenik and Wagner, 2019) means it is difficult to directly use persistence diagrams for machine learning. In order to better integrate diagrams into machine learning workflows, efforts have been made to map them into a more manageable form; primarily through embeddings into finite feature vectors (Kališnik, 2018; Fabio and Ferri, 2015; Chepushtanova et al., 2015), functional summaries (Bubenik, 2015; Rieck et al., 2019), or by defining a positive-definite kernel on diagram space (Reininghaus et al., 2015; Carrière et al., 2017; Le and Yamada, 2018). In all cases this embeds diagrams into a Hilbert space, deforming the metric structure and potentially losing important information (Wagner, 2019; Carrière and Bauer, 2019). These vectorisations have been integrated into deep learning either by learning parameters for the embedding (Hofer et al., 2017; Carrière et al., 2020; Kim et al., 2020; Zhao and Wang, 2019), or as part of a topological loss or regulariser (Chen et al., 2018; Gabrielsson et al., 2020; Clough et al., 2020; Moor et al., 2019). However, the latter technique requires prior knowledge about the correct topology of the dataset, which is clearly not feasible in most scenarios.

Against this background, we give an algorithm to perform Fuzzy c-Means (FCM) clustering (Bezdek, 1980) *directly* on collections of persistence diagrams, enabling learning from persistence diagrams without prior knowledge of the topology or deformation of the metric structure. We perform the convergence analysis for our algorithm, giving the same guarantees as traditional FCM clustering: every convergent subsequence of iterates tends to a local minimum or saddle point. We apply our

Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

technique in two settings: lattice structures in materials science and the decision boundaries of learnt models. A key property for machine learning in materials science has been identified as "invariance to the basis symmetries of physics... rotation, reflection, translation" (Schmidt et al., 2019). Geometric clustering algorithms do not have this invariance, but persistence diagrams do, making them ideally suited for this application; we can cluster transformed lattice structure datasets where geometric equivalents fail. In addition to this, our probabilistic membership values allow us to rank the top-k most likely lattices assigned to a cluster. This is particularly important in materials science, as further investigation requires expensive laboratory time and expertise. Our second application clusters the persistence diagrams of learnt decision boundaries. We use our algorithm to cluster models and tasks, showing that we are able to successfully cluster models to the correct task, and that higher fuzzy membership values imply better performance on unseen tasks.

1.1 Related work

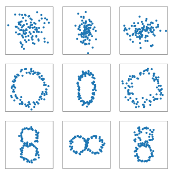
Our work relies on the existence of statistics in the space of persistence diagrams. Mileyko et al. (2011) first showed that means and expectations are well-defined in the space of persistence diagrams. Turner et al. (2012) then developed an algorithm to compute the Fréchet mean. Lacombe et al. (2018) framed the computation of means and barycentres in the space of persistence diagram as an optimal transport problem, allowing them to use the Sinkhorn algorithm (Cuturi and Doucet, 2014) for fast computation of approximate solutions. Maroulas et al. (2017) gave an algorithm for hard clustering persistence diagrams based on Turner et al.'s algorithm. In comparison to hard clustering, our fuzzy algorithm allows probablistic top-k ranking in situations where verifying correctness is expensive, and provides information about proximity to all clusters which we can exploit. We demonstrate these advantages in the experiments. Wasserstein barycentre clustering (WBC) also offers similarities to our algorithm; it clusters datasets of point clouds by the Wasserstein distance between the point clouds, rather than the Wasserstein distance between their persistence diagrams. We compare our algorithm experimentally to WBC using ADMM (Ye and Li, 2014), Bregman ADMM (Ye et al., 2017), Subgradient Descent (Cuturi and Doucet, 2014), Iterative Bregman Projection (Benamou et al., 2015), and full linear programming (Li and Wang, 2008). Each of these algorithms computes or approximates the Wasserstein barycentre in different ways. Theoretically, fuzzy discrete distribution clustering (d. A. T. de Carvalho et al., 2015) is similar to our algorithm, but the addition of the diagonal in the persistence diagram makes our work distinct.

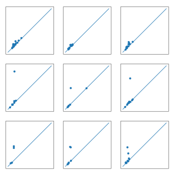
1.2 Our contributions

Our main contribution is an algorithm for Fuzzy c-Means clustering of persistence diagrams, along with the convergence analysis. We show that every convergent subsequence of iterates of our algorithm tends to a local minimum or saddle point of the cost function. This is the same convergence guarantee provided by traditional FCM clustering (Bezdek et al., 1987). Updating the cluster centres requires computing the weighted Fréchet mean. We extend the algorithm given by Turner et al. (2012) to the weighted case, justifying our addition of weights by extending their proof to show that the updated algorithm converges. We implement our algorithm in Python, available in the supplementary materials. It works with persistence diagrams from commonly used open-source libraries for Topological Data Analysis (Dionysus and Ripser) so is available for easy integration into current workflows, offering a powerful unsupervised learning algorithm to data science practitioners using TDA. We demonstrate the application of our algorithm to lattice structures from materials science and decision boundaries. Our algorithm classifies lattice structures where geometric equivalents fail, whilst giving probabilistic rankings to help prioritise expensive further investigation. We also cluster the persistence diagrams of decision boundaries, showing that our fuzzy clustering captures information about model performance on unseen tasks.

2 Background

A persistence diagram is a multiset in the extended plane that concisely represents the topology of a dataset. See Edelsbrunner and Harer (2010) for an introduction to computational topology. We utilise filtered complexes by Barannikov (1994). We use the 2-Wasserstein L_2 metric on the space of persistence diagrams, as it is stable on persistence diagrams of finite point clouds (Chazal et al., 2012). Given diagrams $\mathbb{D}_1, \mathbb{D}_2$, the distance is $W_2(\mathbb{D}_1, \mathbb{D}_2) = \left(\inf_{\gamma:\mathbb{D}_1 \to \mathbb{D}_2} \sum_{x \in \mathbb{D}_1} ||x - \gamma(x)||_2^2\right)^{1/2}$, where the


(a) Synthetic datasets


(b) 1-PH persistence diagrams

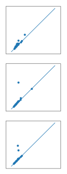

(c) Cluster centres

Figure 1: Our algorithm successfully clustered the datasets by their topology.

infimum is taken over all bijections $\gamma : \mathbb{D}_1 \to \mathbb{D}_2$. We work in $\mathcal{D}_{L^2} = \{\mathbb{D} : W_2(\mathbb{D}, \Delta) < \infty\}$, a complete and separable metric space (Mileyko et al., 2011).

3 Algorithmic design

Given a collection of persistence diagrams $\{\mathbb{D}_j\}_{j=1}^n \subset \mathcal{D}_{L^2}$ and a fixed number of clusters c, we wish to find cluster centres $\{\mathbb{M}_k\}_{k=1}^c \subset \mathcal{D}_{L^2}$, along with membership values $r_{jk} \in [0, 1]$ that denote the extent to which $\mathbb{D}_j$ is associated with cluster $\mathbb{M}_k$. We follow probabilistic fuzzy clustering, so that $\sum_k r_{jk} = 1$ for each j. We extend the FCM algorithm originally proposed by Bezdek (1980). Our r_{jk} is the same as traditional FCM clustering, adapted with the Wasserstein distance. That is,

$$r_{jk} = \left(\sum_{l=1}^{c} \frac{W_2(\mathbb{M}_k, \mathbb{D}_j)}{W_2(\mathbb{M}_l, \mathbb{D}_j)} \right)^{-1}.$$

To update $\mathbb{M}_k$, we compute the weighted Fréchet mean of the persistence diagrams $\{\mathbb{D}_j\}_{j=1}^n$ with the weights $\{r_{jk}^2\}_{j=1}^n$. Specifically, $\mathbb{M}_k = \arg\min_{\hat{\mathbb{D}}} \sum_{j=1}^n r_{jk}^2 W_2(\hat{\mathbb{D}}, \mathbb{D}_j)^2$, for $k = 1, \ldots, c$. By alternatively updating cluster centres and membership values we get a sequence of iterates. We show in Appendix A that every convergent subsequence of these iterates tends to a local minimum or saddle point. To compute the weighted Fréchet mean we extend the algorithm for the unweighted case proposed by Turner et al. (2012). We also extend their proof of convergence, justifying the addition of the weights. The details of this work are in Appendix B.

4 Experiments

4.1 Synthetic data

We demonstrate our algorithm on a synthetic dataset designed to highlight its ability to cluster based on the topology of the underlying datasets. We produce nine datasets, each with either zero, one, or two holes and compute the corresponding 1-PH persistence diagrams. Our algorithm successfully clusters the diagrams based on their topology: Figure 1 shows that the cluster centres have zero, one, or two off-diagonal points. Because we are reducing the cardinality and dimensionality of datasets by mapping into persistence diagrams, we also demonstrate a speed-up of at least an order of magnitude over Wasserstein barycentre clustering methods. Details of these experiments are in Appendix C.1.

4.2 Lattice structures

A key property for machine learning in materials science has been identified as "invariance to the basis symmetries of physics... rotation, reflection, translation" (Schmidt et al., 2019). Removing the need for a standardised coordinate system allows machine learning methods to be applied to a broader range of existing coordinate datasets. Persistence diagrams, which capture affine transformation-invariant properties of datasets, are ideally suited for application in this domain. Additionally, the fuzzy membership values allow top-k ranking of candidates suggested by our algorithm. This is particularly important in materials science, where further investigation of materials can be extremely costly. We apply our algorithm to two examples of lattice structures from materials science: cubic structures and carbon allotropes. The most common lattice mettalic structures are face-centred

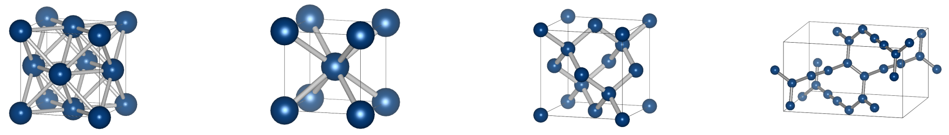


Figure 2: From left to right: FCC and BCC cubic structures, and diamond and cis-hinger polydiacetylene. Our algorithm clusters transformed lattice structures where WBC algorithms fails.

cubic (FCC) structures and body-centred cubic (BCC) structures (Putnis, 1992). Carbon allotropes are important because they are widely anticipated to revolutionise electronics and optoelectronics (Wang et al., 2016). We use atomic positions for the unit-cells of iron mp-150 and iron mp-13 from the Materials Project (Jain et al., 2013), representing BCC and FCC structures respectively, for our first experiment. For our second experiment we use diamond and cis-hinged polydiacetylene unit-cell atomic positions from the Samara Carbon Allotrope Database (Hoffmann et al., 2016). These are shown in Figure 2. We simulate distinct collections of lattices by transforming the atomic coordinates, with no information about bonds given to the algorithms. We successfully cluster the atomic coordinates of the lattice structures regardless of transformations applied. In comparison, we run Wasserstein barycentre clustering on the same datasets using several state-of-the-art algorithms for barycentre computation and approximation. Each can only successfully cluster the cubic structures after reflection, and none of them successfully cluster the carbon allotropes after any transformations. We give specifics in Appendix C.2.

4.3 Decision boundaries

Learnt models have been shown to perform better on datasets which have a similar persistence diagram to the model's decision boundary (Ramamurthy et al., 2019). In fact, topological complexity has been shown to correlate with generalisation ability (Guss and Salakhutdinov, 2018; Gabrielsson and Carlsson, 2019). We utilise our algorithm to cluster the topology of models and tasks, showing that high membership values imply better performance on tasks. Specifically, given a dataset with n classes, we fix one class to define $n-1$ binary classification *tasks*. On each of these tasks, we train a *model*. We compute the decision boundary of the model f, defined as $(x_1, \dots, x_m, f(x))$ where $f(x)$ is the model's prediction for $x = (x_i)_i$, and the decision boundary of the tasks, defined via the labelled dataset as $(x_1, \dots, x_m, y)$ where y is the true label. We compute the 1-persistence diagrams of the tasks' and models' decision boundaries and cluster them to obtain membership values and cluster centres. To view task and model proximity through our clustering, we find the cluster centre with the highest membership value for each task, and consider the models closest to that cluster centre. Note that in general you cannot do this with hard clustering: most of the time a path will not exist from task to cluster centre to model, because each task/model is only associated with a single cluster. This contrasts with fuzzy clustering, where you have information about how close each model/task is to each cluster centre. We further discuss why this does not work for hard clustering in Appendix C.3.

Table 1: Performance change over baseline when using fuzzy membership values for model selection.

	Top-3	Top-2	Top-1
MNIST	$+6.17^{\pm 2.18}$	$+10.81^{\pm 1.88}$	$+20.88^{\pm 4.08}$
FashionMNIST	$+16.46^{\pm 4.00}$	$+21.94^{\pm 4.73}$	$+23.30^{\pm 8.72}$
Kuzushiji	$+6.61^{\pm 1.78}$	$+11.18^{\pm 2.45}$	$+21.89^{\pm 5.54}$

To assess the ability of our model/task clustering, we performed the above experiment on MNIST (LeCun et al., 2010), FashionMNIST (Xiao et al., 2017), and Kuzushiji-MNIST (Clanuwat et al., 2018). Our goal is to evaluate whether or not the clustering is capturing information about model performance on tasks, so as a baseline we use the average performance of models on a fixed task. We start by verifying what happens if we use the model nearest a task (i.e., top-1). We see a significant increase in performance, indicating that the topological fuzzy clustering has selected the model trained on the task. We also compute the top-3 and top-2 performance change over average. We still see a statistically significant increase in performance, indicating that our membership values are capturing information about model performance on unseen tasks (Table 1).

5 Conclusion

We have developed FCM clustering on the space of persistence diagrams, giving an important class of unsupervised learning algorithm to data science practitioners using TDA.

References

Serguei Barannikov. The framed morse complex and its invariants. *Advances in Soviet Mathematics*, 21:93–115, 02 1994.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 2015. doi: 10.1137/141000439.

J. C. Bezdek. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(1):1–8, Jan 1980. ISSN 1939-3539. doi: 10.1109/TPAMI.1980.4766964.

J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker. Convergence theory for fuzzy c-means: Counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5): 873–877, 1987.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3):77–102, 2015. URL `http://jmlr.org/papers/v16/bubenik15a.html`.

Peter Bubenik and Alexander Wagner. Embeddings of persistence diagrams into hilbert spaces. *CoRR*, abs/1905.05604, 2019. URL `http://arxiv.org/abs/1905.05604`.

Mathieu Carrière and Ulrich Bauer. On the metric distortion of embedding persistence diagrams into separable hilbert spaces. In *Symposium on Computational Geometry*, 2019.

Mathieu Carrière, Marco Cuturi, and S. Oudot. Sliced wasserstein kernel for persistence diagrams. In *ICML*, 2017.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, T. Lacombe, Martin Royer, and Y. Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *AISTATS*, 2020.

Frédéric Chazal, Vin Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. 07 2012. doi: 10.1007/978-3-319-42545-0.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. Toporeg: A topological regularizer for classifiers. *CoRR*, abs/1806.10714, 2018. URL `http://arxiv.org/abs/1806.10714`.

Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier. Persistence images: An alternative persistent homology representation. 07 2015.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

J. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China, 22–24 Jun 2014. PMLR. URL `http://proceedings.mlr.press/v32/cuturi14.html`.

F. d. A. T. de Carvalho, A. Irpino, and R. Verde. Fuzzy clustering of distribution-valued data using an adaptive l2 wasserstein distance. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2015.

Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010. ISBN 978-0-8218-4925-5.

Barbara Di Fabio and Massimo Ferri. Comparing persistence diagrams through complex vectors. In *Image Analysis and Processing — ICIAP 2015*, pages 294–305. Springer International Publishing, 2015. doi: 10.1007/978-3-319-23231-7_27. URL `https://doi.org/10.1007/978-3-319-23231-7_27`.

Rickard Brüel Gabrielsson and G. Carlsson. Exposition and interpretation of the topology of neural networks. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1069–1076, 2019.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. volume 108 of *Proceedings of Machine Learning Research*, pages 1553–1563, Online, 26–28 Aug 2020. PMLR. URL `http://proceedings.mlr.press/v108/gabrielsson20a.html`.

William H. Guss and R. Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *ArXiv*, abs/1802.04443, 2018.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1634–1644. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6761-deep-learning-with-topological-signatures.pdf`.

Roald Hoffmann, Artem Kabanov, Andrey Golov, and Davide Proserpio. Homo citans and carbon allotropes: For an ethics of citation. *Angewandte Chemie International Edition*, 55, 07 2016. doi: 10.1002/anie.201600655.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin a. Persson. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. ISSN 2166532X. doi: 10.1063/1.4812323. URL `http://link.aip.org/link/AMPADS/v1/i1/p011002/s1&Agg=doi`.

Sara Kališnik. Tropical coordinates on the space of persistence barcodes. *Foundations of Computational Mathematics*, 19(1):101–129, January 2018. doi: 10.1007/s10208-018-9379-y. URL `https://doi.org/10.1007/s10208-018-9379-y`.

Kwangho Kim, Jisu Kim, J. Kim, Frédéric Chazal, and L. Wasserman. Efficient topological layer based on persistent landscapes. *ArXiv*, abs/2002.02778, 2020.

Theo Lacombe, Marco Cuturi, and Steve Oudot. Large scale computation of means and clusters for persistence diagrams using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9770–9780. Curran Associates, Inc., 2018.

Tam Le and Makoto Yamada. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. In *NeurIPS*, 2018.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

Vasileios Maroulas, Joshua Mike, and Andrew Marchese. K-means clustering on the space of persistence diagrams. In *SPIE*, page 29, 08 2017. doi: 10.1117/12.2273067.

Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems - INVERSE PROBL*, 27, 12 2011. doi: 10.1088/0266-5611/27/12/124007.

M. Moor, Max Horn, Bastian Alexander Rieck, and K. Borgwardt. Topological autoencoders. *ArXiv*, abs/1906.00722, 2019.

Andrew Putnis. *An Introduction to Mineral Sciences*. Cambridge University Press, 1992. doi: 10.1017/CBO9781139170383.

Karthikeyan Natesan Ramamurthy, Kush Varshney, and Krishnan Mody. Topological data analysis of decision boundaries with application to model selection. volume 97 of *Proceedings of Machine Learning Research*, pages 5351–5360, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/ramamurthy19a.html`.

J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4741–4748, 2015.

Bastian Alexander Rieck, F. Sadlo, and H. Leitte. Topological machine learning with persistence indicator functions. *ArXiv*, abs/1907.13496, 2019.

Jonathan Schmidt, Mário R. G. Marques, Silvana Botti, and Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83, 2019. ISSN 2057-3960. doi: 10.1038/s41524-019-0221-0. URL `https://doi.org/10.1038/s41524-019-0221-0`.

Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52:44–70, 2012.

Alexander Wagner. Nonembeddability of persistence diagrams with p>2 wasserstein metric. *ArXiv*, abs/1910.13935, 2019.

Zhanyu Wang, F. Dong, Bo Shen, R. Zhang, Y. Zheng, L. Chen, Songyou Wang, Chongmin Wang, K. Ho, Yuan-Jia Fan, Bih-Yaw Jin, and Wan-Sheng Su. Electronic and optical properties of novel carbon allotropes. *Carbon*, 101:77–85, 01 2016. doi: 10.1016/j.carbon.2016.01.078.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

J. Ye and J. Li. Scaling up discrete distribution clustering using admm. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5267–5271, 2014.

Jianbo Ye, Panruo Wu, James Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, PP:1–1, 01 2017. doi: 10.1109/TSP.2017.2659647.

W.I. Zangwill. *Nonlinear programming: a unified approach*. Prentice-Hall international series in management. Prentice-Hall, 1969. URL `https://books.google.co.uk/books?id=TWhxLcApH9sC`.

Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9859–9870. Curran Associates, Inc., 2019.

A Convergence of the FCM clustering algorithm

We first need to consider our separate updates as a single update procedure. Let $F : \mathbb{M} \mapsto R$ be defined by (1) and $G : R \mapsto \mathbb{M}$ be defined by (2), and for $R = \{r_{jk}\}$ and $\mathbb{M} = \{\mathbb{M}_k\}$ consider the sequence

$$\left\{T^{(l)}(R, \mathbb{M}) : l = 0, 1, \dots\right\}, \text{ where } T(R, M) = (F \circ G(R), G(R)).$$

We wish to show convergence of the iterates of T to a local minimum or saddle point of the cost function

$$J(R, \mathbb{M}) = \sum_{j=1}^{n} \sum_{k=1}^{c} r_{jk}^2 W_2(\mathbb{M}_k, \mathbb{D}_j)^2.$$

The two stage update process of T is too complicated to use standard fixed point theorems, so as in Bezdek (1980) we shall use the following result, which is proven in Zangwill (1969).

Theorem 1 (Zangwill's Convergence Theorem)**.** *Let $A : X \to 2^X$ be a point-to-set algorithm acting on X. Given $x_0 \in X$, generate a sequence $\{x_k\}_{k=1}^{\infty}$ such that $x_{k+1} \in A(x_k)$ for every k. Let $\Gamma \subset X$ be a solution set, and suppose that the following hold.*

(i) The sequence $\{x_k\} \subset S \subset X$ for a compact set S.

(ii) There exists a continuous function Z on X such that if $x \notin \Gamma$ then $Z(y) < Z(x)$ for all $y \in A(x)$, and if $x \in \Gamma$ then $Z(y) \leq Z(x)$ for all $y \in A(x)$. The function Z is called a descent function.

(iii) The algorithm A is closed on $X \setminus \Gamma$.

Then every convergent subsequence of $\{x_k\}$ tends to a point in the solution set Γ.

Our algorithm is the update function T. We define our solution set as

$$\Gamma = \left\{(R, \mathbb{M}) : J(R, \mathbb{M}) < J(\hat{R}, \hat{\mathbb{M}}) \; \forall \; (\hat{R}, \hat{\mathbb{M}}) \in B((R, \mathbb{M}), r)\right\}$$

for some $r > 0$, where the ball surrounding R is the Euclidean ball in $\mathbb{R}^{nc}$ and the ball surrounding $\mathbb{M}$ is $\cup_{k=1}^{c} B_{W_2}(\mathbb{M}_k, r)$. This set contains the local minima and saddle points of the cost function (Bezdek et al., 1987). We wish to show that our cost function $J(R, \mathbb{M})$ is the descent function Z. We proceed by verifying each of the requirements for Zangwill's Convergence Theorem.

Lemma 2. *Every iterate $T^{(l)}(R, \mathbb{M}) \in [0, 1]^{nc} \times \operatorname{conv}(\mathbb{D})^c$, where*

$$\operatorname{conv}(\mathbb{D}) = \bigcup_{k=1}^{c} \bigcup_{\gamma_j} \bigcup_{i=1}^{m} \operatorname{conv}\{\gamma_j(y^{(i)}) : j = 1, \dots, n\},$$

with γ_j a bijection $\mathbb{M}_k \to \mathbb{D}_j$ and $\operatorname{conv}\{\gamma_j(y^{(i)}) : j = 1, \dots, n\}$ the ordinary convex hull in the plane. Furthermore, $[0, 1]^{nc} \times \operatorname{conv}(\mathbb{D})^c$ is compact.

Proof. By construction, every $r_{jk} \in [0, 1]$. Since $j = 1, \dots, n$ and $k = 1, \dots, c$, we can view R as a point in $[0, 1]^{nc}$, and so every iterate of R is in $[0, 1]^{nc}$. We shall show that for a fixed k and a fixed bijection $\gamma_j : \mathbb{M}_k \to \mathbb{D}_j$, each updated $y^{(i)}$ is contained in a convex combination of $\{\gamma_j(y^{(i)}) : j = 1, \dots, n\}$. Where $\gamma_j(y^{(i)}) = \Delta$, let $\gamma_j(y^{(i)}) = w_\Delta$ as defined in (4), as this is the update point we use for the diagonal. Since there are a finite number of off-diagonal points, each updated $\mathbb{M}_k$ is therefore contained in the union over all bijections and all points $y^{(i)}$ of the convex combination of $\{\gamma_j(y^{(i)}) : j = 1, \dots, n\}$. By also taking the union over each k, we show that every iterate of $\mathbb{M}$ must be contained in the finite triple-union of the convex combination of each possible bijection. To show that each updated $y^{(i)}$ is contained in a convex combination of $\{\gamma_j(y^{(i)}) : j = 1, \dots, n\}$, recall that $y^{(i)} = \left(\sum_{j=1}^{n} r_{jk}^2\right)^{-1} \sum_{j=1}^{n} r_{jk}^2 \gamma_j(y^{(i)})$. Letting $t_j^{(i)} = r_{jk}^2 \left(\sum_{j=1}^{n} r_{jk}^2\right)^{-1}$, clearly each $t_j^{(i)} > 0$ and $\sum_{j=1}^{n} t_j^{(i)} = 1$. Since $y^{(i)} = \sum_{j=1}^{n} t_j^{(i)} \gamma_j(y^{(i)})$, each $y^{(i)}$ is contained in the convex combination. Therefore $T^{(l)}(R, \mathbb{M}) \in [0, 1]^{nc} \times \operatorname{conv}(\mathbb{D})^c$ for each $l = 0, 1, \dots$.

Now, $[0, 1]$ is closed and bounded, so is compact. The convex hull of points in the plane is closed and bounded, so $\operatorname{conv}\{\gamma_j(y^{(i)}) : j = 1, \dots, n\}$ is compact. Since finite unions and finite direct products of compact sets are compact, $[0, 1]^{nc} \times \operatorname{conv}(\mathbb{D})^c$ is also compact. $\square$

Lemma 3. *The cost function $J(R, \mathbb{M})$ is a descent function, as defined in Theorem 1(ii).*

Proof. The cost function J is continuous, as it's a sum, product and composition of continuous functions. Furthermore, we have that for any $(R, \mathbb{M}) \notin \Gamma$,

$$J(T(R, \mathbb{M})) = J(F \circ G(R), G(R)) < J(R, G(R)) < J(R, M),$$

where the first inequality is due to Proposition 1 in Bezdek (1980), and the second inequality comes from the definition of the Fréchet mean. If $(R, \mathbb{M}) \in \Gamma$ then the strict inequalities include equality throughout. □

Theorem 4. *For any $(R, \mathbb{M})$, every convergent subsequence of $\{T^{(l)}(R, \mathbb{M}) : l = 0, 1, \dots\}$ tends to a local minimum or saddle point of the cost function J.*

Proof. We proceed with Zangwill's Convergence Theorem. Our algorithm is the update function T, our solution set is Γ, and our descent function is the cost function $J(R, \mathbb{M})$. By Lemma 2, every iterate is contained within a compact set. By Lemma 3, J is a descent function. Finally, since our function T only maps points in the plane to points in the plane, it is a closed map. The theorem follows by applying Theorem 1. □

Algorithm 1 FPDCluster

```
Input Diagrams $\mathbb{D} = \{\mathbb{D}_j\}_{j=1}^n$, number of clusters $c$, maximum iterations MAXITER
Output Cluster centres $\mathbb{M} = \{\mathbb{M}_k\}_{k=1}^c$, membership values $R = \{r_{jk}\}$

 1: $\mathbb{D} = $ AddDiagonals($\mathbb{D}$)
 2: $\mathbb{M} = $ InitCentres($\mathbb{D}$)
 3: for count in 1..MAXITER do
 4:     for $j$ in $1..n$ do
 5:         for $k$ in $1..c$ do
 6:             $r_{jk} \leftarrow \left(\sum_{l=1}^c \frac{W_2(\mathbb{M}_k, \mathbb{D}_j)}{W_2(\mathbb{M}_l, \mathbb{D}_j)}\right)^{-1}$
 7:         end for
 8:     end for
 9:     for $k$ in $1..c$ do
10:         $\mathbb{M}_k \leftarrow$ WFrechetMean($\mathbb{D}, R_k$)
11:     end for
12: end for
13: return $\mathbb{M}, R$
```

B Computing the weighted Fréchet mean

Turner et al. (2012) give an algorithm for the computation of Fréchet means. In this section we extend their algorithm and proof of convergence to the weighted case. In Algorithm 1 we add copies of the diagonal to ensure that each diagram has the same cardinality; denote this cardinality as m. To compute the weighted Fréchet mean, we need to find $\mathbb{M}_k = \{y^{(i)}\}_{i=1}^m$ that minimises the Fréchet function when we update our cluster centre. Implicit to the Wasserstein distance is a bijection $\gamma_j : y^{(i)} \mapsto x_j^{(i)}$ for each j. Supposing we know these bijections, we can rearrange the Fréchet function into the form $F(\mathbb{M}_k) = \sum_{j=1}^n r_{jk}^2 W_2(\mathbb{M}_k, \mathbb{D}_j)^2 = \sum_{i=1}^m \sum_{j=1}^n r_{jk}^2 ||y^{(i)} - x_j^{(i)}||^2$.

In this form, the summand is minimised for $y^{(i)}$ by the weighted Euclidean centroid of the points $\{x_j^{(i)}\}_{j=1}^n$. Therefore to compute the weighted Fréchet mean, we need to find the correct bijections. We start by using the Hungarian algorithm to find an optimal matching between $\mathbb{M}_k$ and each $\mathbb{D}_j$. Given a $\mathbb{D}_j$, for each point $y^{(i)} \in \mathbb{M}_k$, the Hungarian algorithm will assign an optimally matched point $x_j^{(i)} \in \mathbb{D}_j$. Specifically, we find matched points

$$\left[x_j^{(i)}\right]_{i=1}^m \longleftarrow \text{Hungarian}(\mathbb{M}_k, \mathbb{D}_j), \text{ for each } j = 1, \dots, n. \tag{1}$$

Now, for each $y^{(i)} \in \mathbb{M}_k$ we need to find the weighted average of the matched points $\left[x_j^{(i)}\right]_{j=1}^n$. However, some of these points could be copies of the diagonal, so we need to consider three distinct cases: that each matched point is off-diagonal, that each one is a copy of the diagonal, or that the points are a mixture of both. We start by partitioning $1, \ldots, n$ into the indices of the off-diagonal points $\mathcal{J}_{\text{OD}}^{(i)} = \left\{j : x_j^{(i)} \neq \Delta\right\}$ and the indices of the diagonal points $\mathcal{J}_{\text{D}}^{(i)} = \left\{j : x_j^{(i)} = \Delta\right\}$ for each $i = 1, \ldots, m$. Now, if $\mathcal{I}_{\text{OD}} = \emptyset$ then $y^{(i)}$ is a copy of the diagonal. If not, let $w = \left(\sum_{j \in \mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2\right)^{-1} \sum_{j \in \mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2 x_j^{(i)}$ be the weighted mean of the off-diagonal points. If $\mathcal{J}_{\text{D}}^{(i)} = \emptyset$, then $y^{(i)} = w$. Otherwise, let w_Δ be the point on the diagonal closest to w. Then our update is

$$y^{(i)} \longleftarrow \frac{\sum_{j \in \mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2 x_j^{(i)} + \sum_{j \in \mathcal{J}_{\text{D}}^{(i)}} r_{jk}^2 w_\Delta}{\sum_{j=1}^n r_{jk}^2}, \text{ for } i = 1, \ldots, m. \tag{2}$$

We alternate between (1) and (2) until the matching remains the same. Theorem 5 proves that this algorithm converges to a local minimum of the Fréchet function.

Theorem 5. *Given diagrams $\mathbb{D}_j$, membership values r_{jk}, and the Fréchet function $F(\hat{\mathbb{D}}) = \sum_{j=1}^n r_{jk}^2 W_2(\hat{\mathbb{D}}, \mathbb{D}_j)^2$, then $\mathbb{M}_k = \{y^{(i)}\}_{i=1}^m$ is a local minimum of F if and only if there is a unique optimal pairing from $\mathbb{M}_k$ to each of the $\mathbb{D}_j$ and each $y^{(i)}$ is updated via (4).*

We now prove Theorem 5. Recall that the Fréchet mean is computed by finding the $\arg\min$ of

$$F(\hat{\mathbb{D}}) = \sum_{j=1}^n r_{jk}^2 F_j(\hat{\mathbb{D}}), \text{ with } F_j(\hat{\mathbb{D}}) = W_2(\hat{\mathbb{D}}, \mathbb{D}_j)^2, \tag{3}$$

for fixed k. We start by recounting work in Turner et al. (2012), which this section adapts for the weighted Fréchet mean.[1] The proofs we're adapting use a gradient descent technique to prove local convergence. In order to use their techniques, we need to define a differential structure on the space of persistence diagrams.

By Theorem 2.5 in Turner et al. (2012), the space of persistence diagrams $\mathcal{D}_{L^2} = \{\mathbb{D} : W_2(\mathbb{D}, \Delta) < \infty\}$ is a non-negatively curved Alexandrov space. An optimal bijection $\gamma : \mathbb{D}_1 \to \mathbb{D}_2$ induces a unit-speed geodesic $\phi(t) = \{(1-t)x + t\gamma(x) : x \in \mathbb{D}_1, 0 \leq t \leq 1\}$. For a point $\mathbb{D} \in \mathcal{D}_{\text{L}^2}$ we define the tangent cone $T_\mathbb{D}$. Define $\hat{\Sigma}_\mathbb{D}$ as the set of all non-trivial unit-speed geodesics emanating from $\mathbb{D}$. Let $\phi, \eta \in \hat{\Sigma}_\mathbb{D}$ and define the angle between them as

$$\angle_\mathbb{D}(\phi, \eta) = \arccos\left(\lim_{s,t \downarrow 0} \frac{s^2 + t^2 - W_2(\phi(s), \eta(t))^2}{2st}\right) \in [0, \pi]$$

when the limit exists. Then the space of directions $(\Sigma_\mathbb{D}, \angle_\mathbb{D})$ is the completion of $\hat{\Sigma}_\mathbb{D}/\sim$ with respect to $\angle_\mathbb{D}$, with $\phi \sim \eta \iff \angle_\mathbb{D}(\phi, \eta) = 0$. We now define the tangent cone as

$$T_\mathbb{D} = (\Sigma_\mathbb{D} \times [0, \infty))/(\Sigma_\mathbb{D} \times \{0\}).$$

Given $u = (\phi, s), v = (\eta, t)$, we define an inner product on the tangent cone by

$$\langle u, v \rangle = st \cos \angle_\mathbb{D}(\phi, \eta).$$

Now, for $\alpha > 0$ denote the space $(\mathcal{D}_{\text{L}^2}, \alpha W_2)$ as $\alpha\, \mathcal{D}_{\text{L}^2}$ and define the map $i_\alpha : \alpha\, \mathcal{D}_{\text{L}^2} \to \mathcal{D}_{\text{L}^2}$. For an open set $\Omega \subset \mathcal{D}_{\text{L}^2}$ and a function $f : \Omega \to \mathbb{R}$, the differential of f at $\mathbb{D} \in \Omega$ is defined by $d_\mathbb{D} f = \lim_{\alpha \to \infty} \alpha(f \circ i_\mathbb{D} - f(\mathbb{D}))$. Finally, we say that $s \in T_\mathbb{D}$ is a supporting vector of f at $\mathbb{D}$ if $d_\mathbb{D} f(x) \leq -\langle s, x \rangle$ for all $x \in T_\mathbb{D}$.

Lemma 6. *The following two results are proven in Turner et al. (2012).*

(i) Let $\mathbb{D} \in \mathcal{D}_{\text{L}^2}$. Let $F_j(\hat{\mathbb{D}}) = W_2(\hat{\mathbb{D}}, \mathbb{D}_j)^2$. Then if ϕ is a distance-achieving geodesic from $\mathbb{D}$ to $\hat{D}$, then the tangent vector to ϕ at $\mathbb{D}$ of length $2W_2(\hat{\mathbb{D}}, \mathbb{D})$ is a supporting vector at $\mathbb{D}$ of f.

(ii) If $\mathbb{D}$ is a local minimum of f and s is a supporting vector of f at $\mathbb{D}$, then $s = 0$.

[1]In Turner et al. (2012), the Fréchet mean is defined as the arg min of the Fréchet function $F(\hat{\mathbb{D}}) = \int W_2(\hat{\mathbb{D}}, \mathbb{D}_j)^2 d\rho(\hat{\mathbb{D}})$ with the empirical measure $\rho = n^{-1} \sum_{j=1}^n \delta_{\mathbb{D}_j}$. We are using the empirical measure $\rho = \left(\sum_{j=1}^n r_{jk}^2\right)^{-1} \sum_{j=1}^n r_{jk}^2 \delta_{\mathbb{D}_j}$, but for ease we drop the scalar $\left(\sum_{j=1}^n r_{jk}^2\right)^{-1}$ as it is positive, so doesn't affect the minimum of the function.

If there is a unique optimal matching $\gamma_{\mathbb{D}_1}^{\mathbb{D}_3} : \mathbb{D}_1 \to \mathbb{D}_3$, we say that it is induced by an optimal matching $\gamma_{\mathbb{D}_1}^{\mathbb{D}_2} : \mathbb{D}_1 \to \mathbb{D}_2$ if there exists a unique optimal matching $\gamma_{\mathbb{D}_2}^{\mathbb{D}_3} : \mathbb{D}_2 \to \mathbb{D}_3$ such that $\gamma_{\mathbb{D}_1}^{\mathbb{D}_3} = \gamma_{\mathbb{D}_2}^{\mathbb{D}_3} \circ \gamma_{\mathbb{D}_1}^{\mathbb{D}_2}$. Proposition 3.2 from Turner et al. (2012) states that an optimal matching at a point is also locally optimal. In particular, it states the following.

Lemma 7. *Let $\mathbb{D}_1, \mathbb{D}_2 \in \mathcal{D}_{\mathrm{L}^2}$ such that there is a unique optimal matching from $\mathbb{D}_1$ to $\mathbb{D}_2$. Then there exists an $r > 0$ such that for every $\mathbb{D}_3 \in B_{W_2}(\mathbb{D}_2, r)$, there is a unique optimal pairing from $\mathbb{D}_2$ to $\mathbb{D}_3$ that is induced by the matching from $\mathbb{D}_1$ to $\mathbb{D}_2$.*

The following theorem proves that our algorithm converges to a local minimum of the Fréchet function.

Theorem 8. *Given diagrams $\mathbb{D}_j$, membership values r_{jk}, and the Fréchet function F defined in (3), then $\mathbb{M}_k = \{y^{(i)}\}_{i=1}^m$ is a local minimum of F if and only if there is a unique optimal pairing from $\mathbb{M}_k$ to each of the $\mathbb{D}_j$, denoted γ_j, and each $y^{(i)}$ is updated via (2).*

Proof. First assume that γ_j are optimal pairings from $\mathbb{M}_k$ to each $\mathbb{D}_j$, and let s_j be the vectors in $T_{\mathbb{M}_k}$ that are tangent to the geodesics induced by γ_j and are distance-achieving. Then by Lemma 6(i), each $2s_j$ is a supporting vector for the function F_j. Furthermore, $2\sum_{j=1}^n r_{jk}^2 s_j$ is a supporting vector for F, as for any $\hat{\mathbb{D}}$,

$$\begin{aligned} d_{\mathbb{M}_k} F(\hat{\mathbb{D}}) &= d_{\mathbb{M}_k}\left(\sum_{j=1}^n r_{jk}^2 F_j(\hat{\mathbb{D}})\right) = \sum_{j=1}^n r_{jk}^2 d_{\mathbb{M}_k} F_j(\hat{\mathbb{D}}) \\ &\leq \sum_{j=1}^n -r_{jk}^2 \langle 2s_j, \hat{\mathbb{D}}\rangle = -\left\langle 2\sum_{j=1}^n r_{jk}^2 s_j, \hat{\mathbb{D}}\right\rangle. \end{aligned}$$

By Lemma 6(ii), $2\sum_{j=1}^n r_{jk}^2 s_j = 0$. Putting $s_j = \gamma_j(y^{(i)}) - y^{(i)}$ and rearranging gives that $y^{(i)}$ updates via (4), as required. Note that when $\gamma_j(y^{(i)}) = \Delta$, we let $\gamma_j(y^{(i)}) = w_\Delta$ as defined in (4), because this minimises the transportation cost to the diagonal. Now suppose that γ_j and $\tilde{\gamma}_j$ are both optimal pairings. Then by the above argument $\sum_{j=1}^n r_{jk}^2 s_j = \sum_{j=1}^n r_{jk}^2 \tilde{s}_j = 0$, implying that $s_j = \tilde{s}_j$ and so $\gamma_j = \tilde{\gamma}_j$. Therefore the optimal pairing is unique.

To prove the opposite direction, assume that $\mathbb{M}_k = \{y^{(i)}\}$ locally minimises the Fréchet function F. Observe that for a fixed bijection γ_j, we have that

$$\begin{aligned} F(\mathbb{M}_k) &= \sum_{j=1}^n r_{jk}^2 W_2(\mathbb{M}_k, \mathbb{D}_j)^2 \\ &= \sum_{j=1}^n r_{jk}^2 \left(\inf_{\gamma_j : \hat{M} \to \mathbb{D}_j} \sum_{y \in \mathbb{M}_k} ||y - \gamma_j(y)||^2\right) \\ &= \sum_{j=1}^n r_{jk}^2 \sum_{i=1}^m ||y^{(i)} - x_j^{(i)}||^2 \\ &= \sum_{i=1}^m \left(\sum_{j=1}^n r_{jk}^2 ||y^{(i)} - x_j^{(i)}||^2\right). \end{aligned}$$

The final term in brackets is non-negative, and minimised exactly when $y^{(i)}$ is updated via (2). Furthermore, the unique optimal pairing from $\mathbb{M}_k$ to each of the $\mathbb{D}_j$'s is the same for every $\hat{\mathbb{M}}$ within the ball $B_{W_2}(\mathbb{M}_k, r)$ for some $r > 0$, by Lemma 7. Therefore, if $\mathbb{M}_k$ is a local minimum of F, then the $y^{(i)}$'s are equal to the values found by taking the optimal pairings γ_j and calculating the weighted means of $\gamma_j(y^{(i)})$ with the weights r_{jk}^2, as required. It will remain a minimum as long as the matching stays the same, which happens in the ball $B_{W_2}(\mathbb{M}_k, r)$, so we are done. □

Algorithm 2 WFrechetMean

Input Diagrams $\mathbb{D} = \{\mathbb{D}_j\}_{j=1}^n$, Weights $R_k = \{r_{jk}\}_{j=1}^n$ (fixed k)

Output Weighted Fréchet mean $\mathbb{M}_k = \{y^{(i)}\}_{i=1}^m$

```
1: $m \leftarrow \max_{1\leq j\leq n} |\mathbb{D}_j|$
2: for $j$ in $1..n$ do
3:     $\left[x_j^{(i)}\right]_{i=1}^m \leftarrow$ HUNGARIAN$(\mathbb{M}_k, \mathbb{D}_j)$
4: end for
5: while $\left\{\left[x_j^{(i)}\right]_{i=1}^m\right\}_{j=1}^n \neq \left\{\left[\hat{x}_j^{(i)}\right]_{i=1}^m\right\}_{j=1}^n$ do
6:     for $i$ in $1..m$ do
7:         $\mathcal{J}_{\text{OD}}^{(i)} = \{j : x_j^{(i)} \neq \Delta\}$
8:         $\mathcal{J}_{\text{D}}^{(i)} = \{j : x_j^{(i)} = \Delta\}$
9:         if $\mathcal{J}_{\text{OD}}^{(i)} = \emptyset$ then
10:            $y^{(i)} \leftarrow \Delta$
11:        else
12:            $w = \left(\sum_{j\in\mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2\right)^{-1} \sum_{j\in\mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2 x_j^{(i)}$
13:            if $\mathcal{J}_{\text{D}}^{(i)} = \emptyset$ then
14:                $y^{(i)} \leftarrow w$
15:            else
16:                $y^{(i)} \leftarrow \frac{\sum_{j\in\mathcal{J}_{\text{OD}}^{(i)}} r_{jk}^2 x_j^{(i)} + \sum_{j\in\mathcal{J}_{\text{D}}^{(i)}} r_{jk}^2 w_\Delta}{\sum_{j=1}^n r_{jk}^2}$
17:            end if
18:        end if
19:    end for
20:    $\left\{\left[\hat{x}_j^{(i)}\right]_{i=1}^m\right\}_{j=1}^n \leftarrow \left\{\left[x_j^{(i)}\right]_{i=1}^m\right\}_{j=1}^n$
21:    for $j$ in $1..n$ do
22:        $\left[x_j^{(i)}\right]_{i=1}^m \leftarrow$ HUNGARIAN$(\mathbb{M}_k, \mathbb{D}_j)$
23:    end for
24: end while
25: return $\{y^{(i)}\}_{i=1}^m$
```

C Experimental details

C.1 Synthetic data

Membership values. The membership values for the synthetic datasets are in Table 2. Datasets 1-3 are the datasets of noise, datasets 4-6 are the datasets with one ring, and datasets 7-9 are the datasets with two rings. We ran our algorithm for 20 iterations.

Table 2: Membership values for the synthetic dataset

Dataset	1	2	3	4	5	6	7	8	9
Cluster 1	0.6336	0.5730	0.5205	0.2760	0.2503	0.1974	0.2921	0.2128	0.2292
Cluster 2	0.1768	0.2057	0.2327	0.5361	0.5329	0.6371	0.2452	0.2291	0.1822
Cluster 3	0.1900	0.2212	0.2468	0.1879	0.2169	0.1655	0.4627	0.5580	0.5885

Timing experiments. For the timing experiments we divide the total number of points equally between four distributions, two of which are noise and two of which are shaped in a ring. Each clustering algorithm was run for five iterations on one core of a 2019 MacBook Pro with a 1.4GHz Intel Core i5. We included the time taken to compute the persistence diagrams in the running times for our algorithm.

We also use synthetic data to empirically compare the running time of our algorithm to other dataset clustering algorithms available. Computing the Wasserstein distance has super-cubic time complexity (Ling and Okada, 2007), so is a significant bottleneck both for our algorithm and comparable Wasserstein barycentre clustering algorithms (Benamou et al., 2015; Cuturi and Doucet, 2014; Li and Wang, 2008; Ye and Li, 2014; Ye et al., 2017). Persistence diagrams generally reduce both the

Table 3: Seconds per clustering iteration

Points	100	200	300	400	500	600	700	800	900	1000
FPDCluster	**0.01552**	**0.1975**	**0.9358**	**2.229**	**5.694**	**12.29**	**19.27**	**34.50**	**53.20**	**77.81**
ADMM	5.622	34.86	161.3	617.6	-	-	-	-	-	-
BADMM	0.2020	2.188	26.38	112.6	-	-	-	-	-	-
SubGD	0.4217	2.273	22.17	103.4	-	-	-	-	-	-
IterBP	0.3825	2.226	21.57	108.9	-	-	-	-	-	-
LP	0.3922	2.031	22.32	117.3	-	-	-	-	-	-

dimensionality and number of points in a dataset,[2] so we in turn reduce the computational bottleneck. To demonstrate this, we evaluated the average time per iteration of our persistence diagram clustering algorithm, as well as the average iteration time for comparable Wasserstein barycentre clustering algorithms. We included the time taken to compute the persistence diagrams from the datasets when timing our clustering algorithm. We give the results in Table 3, leaving an entry blank where it became unpractical to run a test (e.g. it takes too long to return a solution and the algorithm becomes unresponsive). We show at least an order of magnitude improvement in performance over comparable Wasserstein barycentre clustering algorithms.

C.2 Lattice structures

The results obtained are in Tables 5-8. The fuzzy values for FPDCluster are given as floats, although in each case they converged to an absolute cluster. We simulate distinct collections of lattices by transforming the atomic coordinates, with no information about bonds given to the algorithms. The Wasserstein barycentre clustering algorithms each have discrete labels. The correct labellings are for 1-3 and 4-6 to be clustered together in each case. We clustered the 2-PH diagrams. We denote a label as having been assigned by 1, or not assigned by 0. We ran each algorithm for five iterations. We obtained our datasets as cif files, converted them to xyz files, and then to csv files, producing a list of the coordinates of each atom in $\mathbb{R}^3$. We create three copies of each structure. For rotation, we rotate two of them by 180^o around different axes. For reflection, we reflect two of them in different axes. For translation, we translate them up or down by the length of the unit-cell. We use our own python implementation of FPDCluster, available in the supplementary materials. For each of the other algorithms, we use the implementation provided at `https://github.com/bobye/WBC_Matlab`, a copy of which is in the supplementary materials. We do not limit the number of points in the diagram when clustering.

C.3 Decision boundaries

Why hard clustering doesn't work. In order to assign each task to the top-ranked models, we need to have a path from a task to the nearest cluster centre, then from that cluster centre to the k-nearest models (note that when we refer to models/tasks, we're implicitly referring to the persistence diagram of their decision boundary). We can always find that route when fuzzy clustering, as the fractional membership values mean that we have information about the proximity of every model/task with every cluster centre. However, with hard clustering we cannot always find that route. Firstly, the hard labelling means that you lose a lot of information about the proximity of models/tasks to cluster centres. Therefore, in order to find a route, we need a every task to be assigned to a cluster centre that also has a model assigned to it. However, there are no guarantees that will happen. We show an example where no path exists in Figure 3.

Experimental details. All code used for computation is available in the supplementary materials. For models, we trained the standard Pytorch CNN available at `https://github.com/pytorch/examples/blob/master/mnist/main.py`. We trained them on MNIST, FashionMNIST, and KMNIST, each obtained using the Torchvision.datasets package. We split the data into 9 binary datasets for classification, class 0 vs each of the remaining classes. We trained three of each model,

[2]Persistence diagrams are always planar, so if the data is in $\mathbb{R}^d, d > 2$, then there is a dimensionality reduction. For $p > 0$, the persistence diagram of p-PH always has less points than the dataset when computed with the Rips complex.

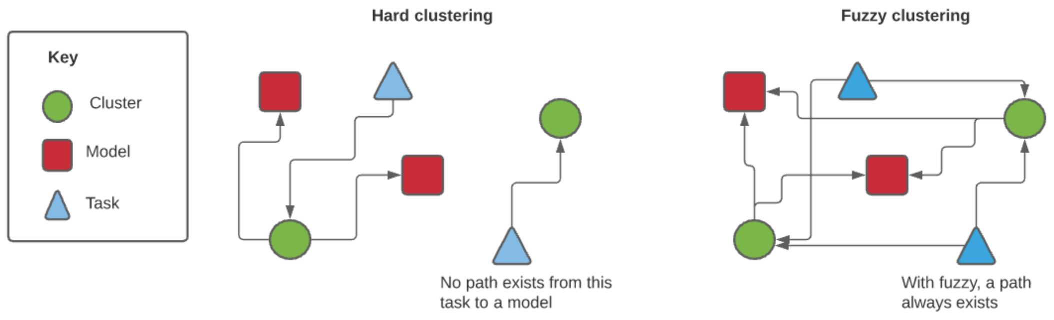

Key
Cluster
Model
Task
Hard clustering
No path exists from this task to a model
Fuzzy clustering
With fuzzy, a path always exists

Figure 3: With hard clustering, we cannot always find a path from a task to a model.

seeded with 0, 1, and 2 respectively. MNIST and KMNIST were each trained for five epochs, FashionMNIST was trained for 14 epochs. We used Ripser to compute the 1-persistence diagrams using default settings. We limited the number of points in the diagram to the 25 most persistent when clustering. Our percentage improvement values use the membership values after 16 iterations. We compute the standard error bounds when calculating the percentage improvement.

Table 4: Clustering results after transformation

	Cubic Structures				Carbon Allotropes			
	None	Rotate	Reflect	Translate	None	Rotate	Reflect	Translate
FPDCluster	✔	✔	✔	✔	✔	✔	✔	✔
ADMM	✓	✗	✓	✗	✓	✗	✗	✗
BADMM	✓	✗	✓	✗	✓	✗	✗	✗
SubGD	✓	✗	✓	✗	✓	✗	✗	✗
IterBP	✓	✗	✓	✗	✓	✗	✗	✗
LP	✓	✗	✓	✗	✓	✗	✗	✗

Table 5: Membership values for non-transformed datasets

		Cubic Structure Datasets						Carbon Allotrope Datasets					
		1	2	3	4	5	6	1	2	3	4	5	6
FPDCluster	Cluster 1	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000
	Cluster 2	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000
ADMM	Cluster 1	1	1	1	0	0	0	1	1	1	0	0	0
	Cluster 2	0	0	0	1	1	1	0	0	0	1	1	1
BADMM	Cluster 1	1	1	1	0	0	0	1	1	1	0	0	0
	Cluster 2	0	0	0	1	1	1	0	0	0	1	1	1
SubGD	Cluster 1	1	1	1	0	0	0	1	1	1	0	0	0
	Cluster 2	0	0	0	1	1	1	0	0	0	1	1	1
IterBP	Cluster 1	1	1	1	0	0	0	1	1	1	0	0	0
	Cluster 2	0	0	0	1	1	1	0	0	0	1	1	1
LP	Cluster 1	1	1	1	0	0	0	1	1	1	0	0	0
	Cluster 2	0	0	0	1	1	1	0	0	0	1	1	1

Table 6: Membership values for rotated datasets

		Cubic Structure Datasets						Carbon Allotrope Datasets					
		1	2	3	4	5	6	1	2	3	4	5	6
FPDCluster	Cluster 1	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000
	Cluster 2	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000
ADMM	Cluster 1	0	0	1	0	0	1	1	0	1	1	0	1
	Cluster 2	1	1	0	1	1	0	0	1	0	0	1	0
BADMM	Cluster 1	0	1	1	0	1	1	1	1	0	1	1	0
	Cluster 2	1	0	0	1	0	0	0	0	1	0	0	1
SubGD	Cluster 1	0	1	1	0	1	1	1	0	0	1	0	0
	Cluster 2	1	0	0	1	0	0	0	1	1	0	1	1
IterBP	Cluster 1	0	1	0	0	1	0	0	1	1	0	1	1
	Cluster 2	1	0	1	1	0	1	1	0	0	1	0	0
LP	Cluster 1	1	0	1	1	0	1	0	1	0	0	1	0
	Cluster 2	0	1	0	0	1	0	1	0	1	1	0	1

Table 7: Membership values for reflected datasets

		Cubic Structure Datasets						Carbon Allotrope Datasets					
		1	2	3	4	5	6	1	2	3	4	5	6
FPDCluster	Cluster 1	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000
	Cluster 2	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000
ADMM	Cluster 1	1	1	1	0	0	0	0	0	0	1	1	0
	Cluster 2	0	0	0	1	1	1	1	1	1	0	0	1
BADMM	Cluster 1	1	1	1	0	0	0	0	0	0	1	1	0
	Cluster 2	0	0	0	1	1	1	1	1	1	0	0	1
SubGD	Cluster 1	1	1	1	0	0	0	1	1	1	1	1	0
	Cluster 2	0	0	0	1	1	1	0	0	0	0	0	1
IterBP	Cluster 1	1	1	1	0	0	0	0	0	0	0	0	1
	Cluster 2	0	0	0	1	1	1	1	1	1	1	1	0
LP	Cluster 1	1	1	1	0	0	0	0	0	0	0	1	1
	Cluster 2	0	0	0	1	1	1	1	1	1	1	0	0

Table 8: Membership values for translated datasets

		Cubic Structure Datasets						Carbon Allotrope Datasets					
		1	2	3	4	5	6	1	2	3	4	5	6
FPDCluster	Cluster 1	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000
	Cluster 2	0.000	0.000	0.000	1.000	1.000	1.000	0.000	0.000	0.000	1.000	1.000	1.000
ADMM	Cluster 1	0	0	1	0	0	1	0	0	1	0	0	1
	Cluster 2	1	1	0	1	1	0	1	1	0	1	1	0
BADMM	Cluster 1	0	1	0	1	1	0	1	1	0	1	1	0
	Cluster 2	1	0	1	0	0	1	0	0	1	0	0	1
SubGD	Cluster 1	0	0	1	0	0	1	0	1	0	0	1	0
	Cluster 2	1	1	0	1	1	0	1	0	1	1	0	1
IterBP	Cluster 1	0	1	0	0	1	0	0	0	1	0	0	1
	Cluster 2	1	0	1	1	0	1	1	1	0	1	1	0
LP	Cluster 1	0	0	1	0	0	1	1	0	1	1	0	1
	Cluster 2	1	1	0	1	1	0	0	1	0	0	1	0