Supplementary Material: Hierarchical Quick Shift Guided Recurrent Clustering

Muzaffer Can Altinigneli
Institut für Informatik
LMU München
Munich, Germany
altinigneli@dbs.ifi.lmu.de

Lukas Miklautz
Faculty of Computer Science
University of Vienna
Vienna, Austria
lukas.miklautz@univie.ac.at

Christian Böhm
Institut für Informatik
MCML, LMU München
Munich, Germany
boehm@dbs.ifi.lmu.de

Claudia Plant
Faculty of Computer Science
ds:UniVie, University of Vienna
Vienna, Austria
claudia.plant@univie.ac.at

I. Introduction

An incomplete taxonomy of different clustering approaches is depicted in Table I.

TABLE I: Categorization of Clustering Methods[a]

	Flat	***Hierarchical***
Parametric	K-Means	Ward
Objective-based	Gaussian Mixture Model	Complete-Linkage
Nonparametric	DBSCAN [1]	HDBSCAN* [2], [3]
Density-based	Mean-Shift, Quick-Shift [4]	**HQuickShift**

[a] HDBSCAN, Fast Density Based Clustering, "the How and the Why" - John Healy, PyData NYC 2018, Published on Feb 2019 (link)

II. Experiments

A. Real Datasets

a) Dimension Reduction: First we express our dimension reduction methodology of datasets. If we inspect the results presented in Fig.9 of [5], we note that DBSCAN performs consistently worse than other methods (including K-Means) especially for relatively high dimensional datasets with low amounts of samples in terms of clustering validation scores: Adjusted Random Index (ARI) and Adjusted Mutual Information (AMI). We believe that such an observation requires a more thoughtful and astute analysis. We claim that validation scores are lower due to the fact that score calculations handle the noise points simply as an extra cluster. One of the features of DBSCAN like HQuickShift is that it is noise aware and it can sort some samples out as *noise* objects.

We work through the *phonemes* dataset (Table II) to demonstrate our claim. With HDBSCAN* method, the best scores that we obtain are ARI: 0.36 and AMI: 0.4. Those scores are evidently lower than the other methods given in Fig.9 of [5]. If we take into account the relative amount of the noise objects, we observe that less than half (nearly 40%) of the data is really clustered and the rest is assigned to noise. If we instead only look at validation scores of the subset of the data (40%) that HDBSCAN* was assigned to clusters, the best scores that we obtain are ARI: 0.96 and AMI: 0.89 for those subset samples. And here we can recognize that where HDBSCAN* could assign samples to clusters confidently, it achieved that nearly perfect clustering with better scores compared to other methods. The main issue is that, as any noise aware density-based clustering algorithm, HDBSCAN* and similarly HQuickShift suffer from the curse of dimensionality, where high dimensional data requires more observed samples to produce density. Therefore, we reduce the dimension of the dataset before we cluster with HQuickShift. We utilize a nonlinear a dimension reduction technique, because the performance of linear methods (e.g. PCA) quickly degrade (e.g. the amount of explained variance) if the target dimension is less than, e.g. ten, in case of *phonemes* dataset. We make use of Uniform Manifold Approximation and Projection (UMAP) [6] to perform nonlinear manifold aware dimension reduction so that the dataset is reduced down to a number of dimensions that is small enough for a noise aware density-based clustering algorithm do not assign majority of samples as noise objects. An alternative method could be adoption of an autoencoder.

b) UMAP Parameter Selection: We select a relatively large $n\_neighbors = 50$ value in UMAP to focus more on global structure rather than on local structure. Focusing on local structure is more prone to introducing small clusters that may be result of noise in the data than real clusters. We set $min\_dist = 0$ to minimum possible value that packs samples together densely and enhances the separations between clusters.

If we select the embedding dimension of UMAP as six and apply UMAP with the parameters given above, the best scores that we obtain are ARI: 0.78 and AMI: 0.81. Furthermore, we cluster over 99% of the data and assigned much less samples as noise objects. Thus, we get better validation scores compared to other methods in Fig.9 of [5], because we no longer have to deal with the relative lack of density in 256 dimensional space. Actually, we increased the confidence of HDBSCAN* by reducing dataset dimensions and clustered more data.

c) Used Datasets and Methods: Banknote, glass, iris, seeds and page-blocks are obtained from UCI repository [7]. Phonemes dataset is originating from [8]. We also use a subset

TABLE II: **Summary of Validation scores for real datasets**: For each dataset, the first row is the adjusted rand index score and the second row is the adjusted mutual information score. The highest scores are illustrated in bold.

Dataset	N	d	n_clust	HQuickShift	Quickshift++	HQuickShift+RNN
banknote	1372	4	2	ARI: 0.800 AMI: 0.632	ARI: 0.690 AMI: 0.555	ARI: **0.951** AMI: **0.910**
glass	214	10	6	ARI: 0.287 AMI: 0.403	ARI: **0.334** AMI: **0.441**	ARI: 0.295 AMI: 0.394
iris	150	4	3	ARI: **0.568** AMI: **0.577**	ARI: **0.568** AMI: **0.577**	ARI: 0.544 AMI: 0.542
mnist	1797	64	10	ARI: 0.752 AMI: 0.812	ARI: **0.815** AMI: **0.840**	ARI: 0.770 AMI: 0.826
seeds	210	7	3	ARI: 0.414 AMI: 0.505	ARI: **0.752** AMI: **0.707**	ARI: 0.403 AMI: 0.473
phonemes	4508	256	5	ARI: **0.760** AMI: **0.780**	ARI: 0.490 AMI: 0.585	ARI: 0.7591 AMI: 0.777
page-blocks	5473	10	5	ARI: **0.577** AMI: **0.354**	ARI: 0.029 AMI: 0.090	ARI: 0.570 AMI: 0.300

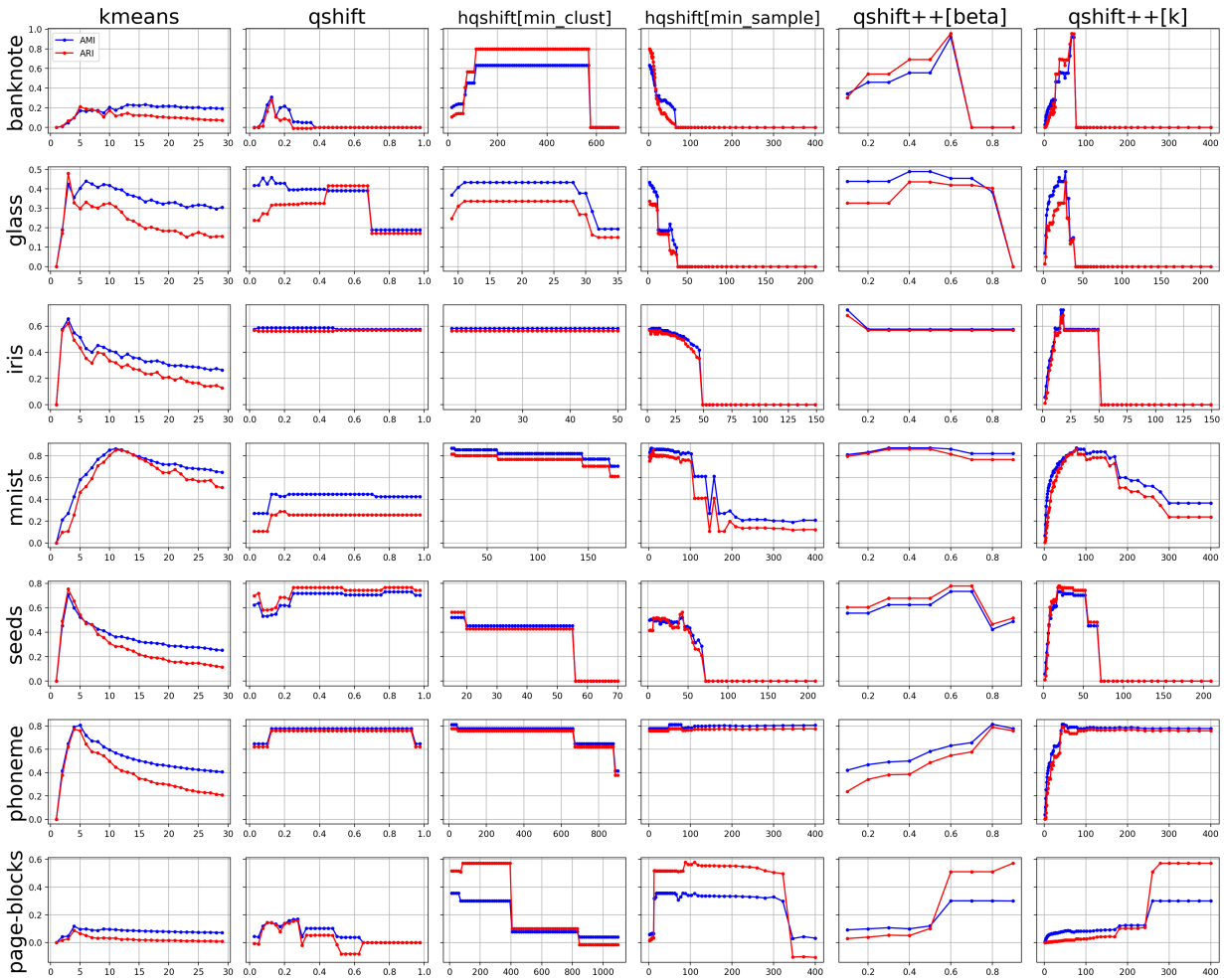

Fig. 1: **Algorithm hyper-parameter setting**: For small min_sample and a broad range of $min\_cluster\_size$, HQuickShift delivers good performance. Selecting k of Quickshift++ is not straightforward since k is related more to the density or sparseness of dataset. **Referred as Figure 8 in paper!**

of mnist UCI dataset from [9] for our experiments.

We evaluate HQuickShift, the RNN trained with the trajectories produced by HQuickShift and Quickshift++ performances on the same datasets after dimension reduction. We reduce all of dataset dimensions to four, i.e. the minimum number of attributes in Table II, without sacrificing the perfor-

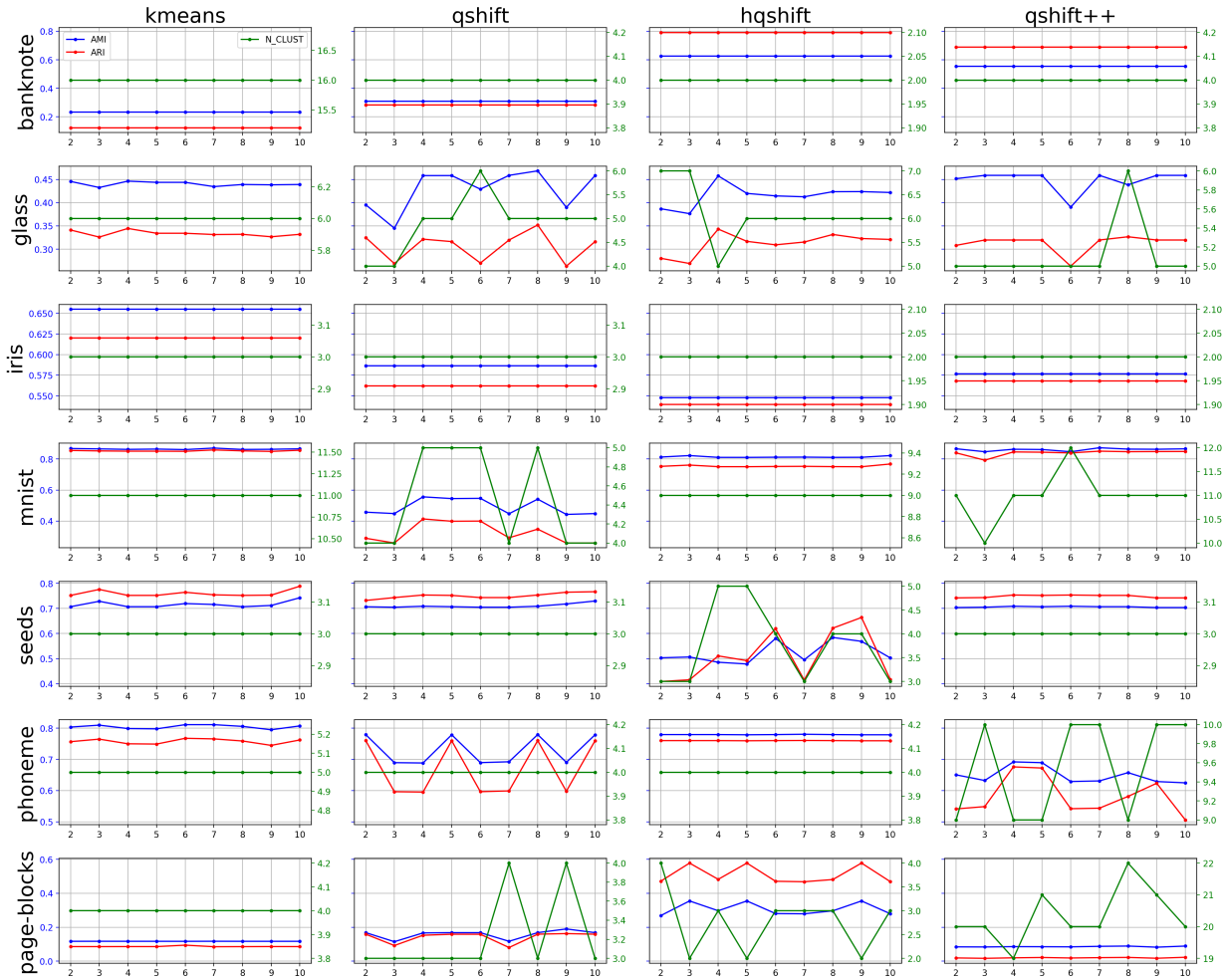

Fig. 2: **Validation score variations due to distinct UMAP-seeds**: Depending on the result of nonlinear dimensionality reduction, the validation scores alter. K-Means is least prone to and none of the remaining methods is immune to variations due to UMAP. **Referred as Figure 9 in paper!**

mance considerably. For nonlinear dimension reduction, we set the UMAP parameters $n\_neighbors = 50$ and $min\_dist = 0$ for all datasets. We present the performance results in terms of ARI and AMI in Table II. These performance results are based on our hyper-parameter analysis depicted in Fig.1. K-Means has a single hyper-parameter, i.e. number of clusters (N_CLUST). QuickShift has two hyper-parameters that is reduced into one by setting $bandwidth = tau$. HQuickShift has three hyper-parameters that is reduced into two by setting $min\_cluster\_size = min\_mode\_size$ and leaving min_sample as-is. Quickshift++ has two hyper-parameters k and β. Since we know the true labels, we can perform such an analysis that is not possible in practice. Our goal is to demonstrate the robustness of the methods with respect to the hyper-parameters used and select a single set of parameters for all datasets for our experiments. For the methods Quickshift++ and HQuickShift with two (remaining) hyper-parameters, we search for the best hyper-parameter index that produces the best AMI score for each hyper-parameter and while keeping it constant at that optimal, we plot the variation of performance values with respect to remaining one. For this reason, we have two columns for both of these methods in Fig.1. We trained ten distinct RNN models for each dataset to produce the validation scores. Since the score variation between models was at the fourth significant digit, we present the mean value of all models in II. We use a two-layer RNN with Gated Recurrent Units [10] and a single set of parameters during RNN training with training and validation samples: $number\_of\_hidden\_neurons = 32$, $batch\_size = 64$, $back\_propagation\_through\_time\_constant = 20$, $learning\_rate = 1e-2$ and $number\_of\_iterrations = 50$.

We obtain fairly satisfactory results with K-Means for some datasets only if we can guess the number of clusters correctly. QuickShift performs reasonable for some datasets pushing also partially the performance of HQuickShift to high, if hierarchical nature of the data does not take the lead. For HQuickShift, it is evident that we can select a very small min_sample size for good performance. At the end, we set

$min\_sample = 3$ for all datasets. Smallest size grouping that is relevant to be eligible a mode (or a cluster) dictates min_mode_size that is naturally an easy parameter to choose during exploratory data analysis. Over a broad range, it stays robust. Depending on the dataset sample size (N), we set a global constant: $min\_mode\_size = 125$ if $N > 1000$ and $min\_mode\_size = 17$ if otherwise. Similarly, we set $\beta = 0.6$ of Quickshift++ that delivers keen performance over the all datasets. But, selecting k is not that straightforward and it influences the performance considerably. k is related to the density rather than N of the dataset. We tried our best and set a global constant: $k = 60$ if $N > 1000$ and $k = 25$ if otherwise. In addition to that, since UMAP is nondeterministic and generates a locally optimal result, we made nine additional runs with different UMAP-seeds and with the same global parameter setting to check the validation performances at distinct runs as shown in Fig.2. Depending on number of clusters (N_CLUST) found and the lower dimensional representation of data, the validation scores may vary. K-Means is least prone and none of the remaining methods is immune to variations due to UMAP.

In general, HQuickShift and HQuickShift+RNN perform close to or outperform Quickshift++. Especially, when HQuickShift has difficulty to cluster dataset due to noise, RNN can enhance the performance by assigning objects to corresponding modes, as in the *banknote* dataset, so called Quick Shift Effect. For small datasets like *seeds*, a single sample may lead to linkage of two close clusters and reduces the validation score. The main advantage of HQuickShift is the ease of parameter selection that is demanding in practice. The parameters min_mode_size and $min\_cluster\_size$ are merely related to the sample size and stay stable over wide parameter range.

REFERENCES

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dl.acm.org/citation.cfm?id=3001460.3001507

[2] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[3] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 5:1–5:51, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2733381

[4] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 705–718.

[5] H. Jiang, J. Jang, and S. Kpotufe, "Quickshift++: Provably good initializations for sample-based mean shift," vol. abs/1805.07909, 2018. [Online]. Available: http://arxiv.org/abs/1805.07909

[6] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv e-prints*, p. arXiv:1802.03426, Feb 2018.

[7] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2009. [Online]. Available: https://web.stanford.edu/~hastie/ElemStatLearn/datasets/phoneme.data

[9] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits

[10] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555