Supplementary: Metric Learning for High-Throughput Combinatorial Data Sets

Kiran Vaddi* and Olga Wodo*

Department of Materials Design and Innovation, Buffalo, NY, USA

E-mail: kiranvad@buffalo.edu; olgawodo@buffalo.edu

Phone: +1 (716)645 1377

Contours of DOFs for synthetic data sets

Figures S1 and S2 shows the DOF variation with in the composition space for the SET A,B respectively.

Hyperparameter selection

In this work, we perform Bayesian optimization (BO) to find hyperparameters. We use a 95% train and 5% validation split of the data.[1] We perform BO by treating the kNN-classification error as the black box function to be optimized.We use the automatic relevance determination squared exponential kernel as a covariance function, and the expected improvement as our acquisition function. We make a total of 15 iterative searches and typically observe about 4 and 15 % training and validation error, respectively.

One limitation that needs to be considered is that the Bayesian optimization is not effective when dealing with a higher number of variables. Such a situation might arise when working with material data sets screened over a large number of design variables.

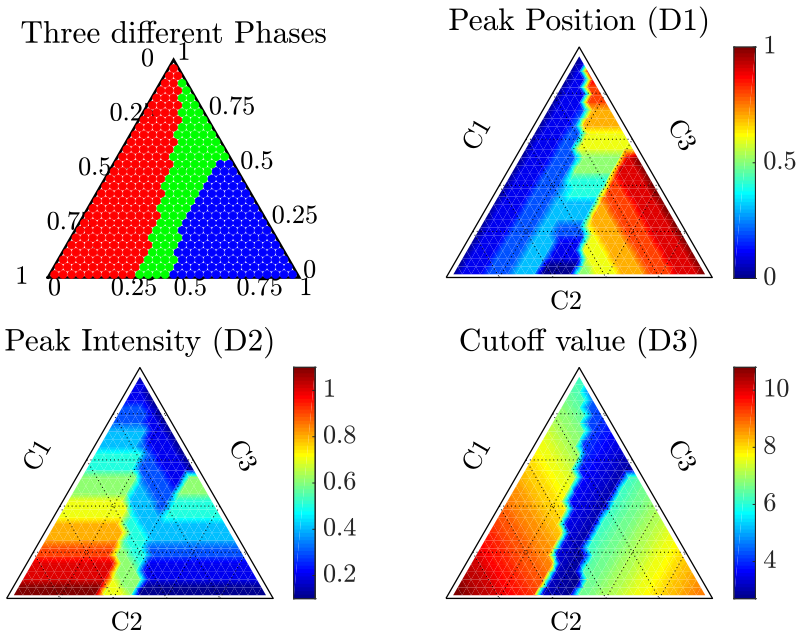


Figure S1: A contour plot of DOFs shows that when the underlying DOFs of the responses are known, it is easier to separate the phases. Although the D1, D2 show overlap in their trends for the three phases, D3 makes it easier to identify phase green to be a separate cluster

For example, high-throughput combinatorial experiments of multi-component systems with various material processing conditions could result in a large number of hyperparameters γ. In such cases, MT-LMNN coupled with BO could be inefficient. Nevertheless, this is an active area of research with promising approached already being proposed.[2]

Defining classes in MT-LMNN

As discussed in the paper, one needs to define classes for each task to emphasize design space information into the metric learned. Here, the user needs to make a decision. Typically the user is interested in quantifying a similarity that explains the low, medium, high contents in each of the design variables giving rise to 3 classes per task.

However, the optimal number of classes for a given task can be identified through optimization. In such a case, partition of composition space can be made into n classes. Parameter n becomes a hyperparameter to partition a data in the sorted composition space.

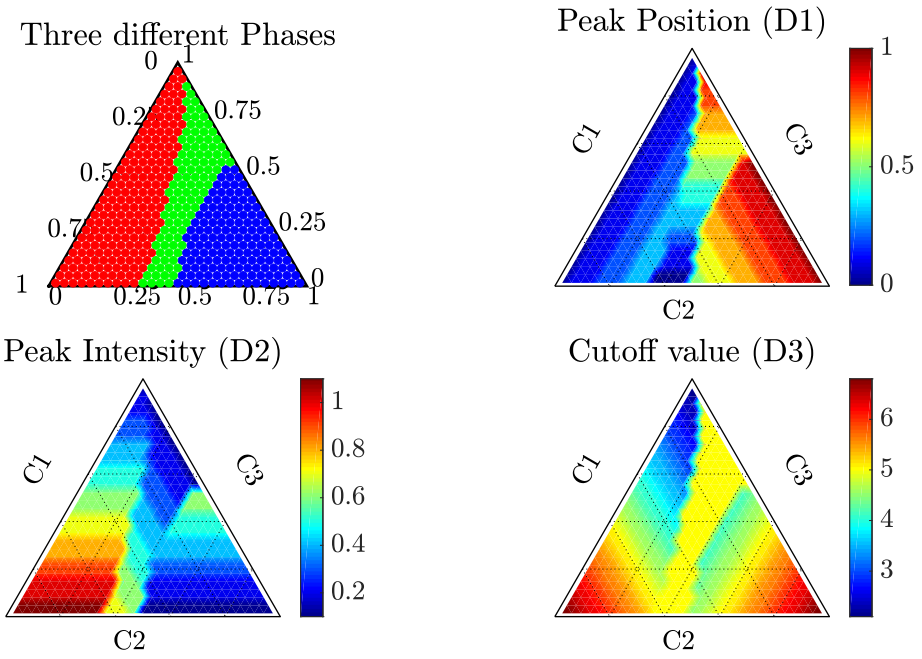
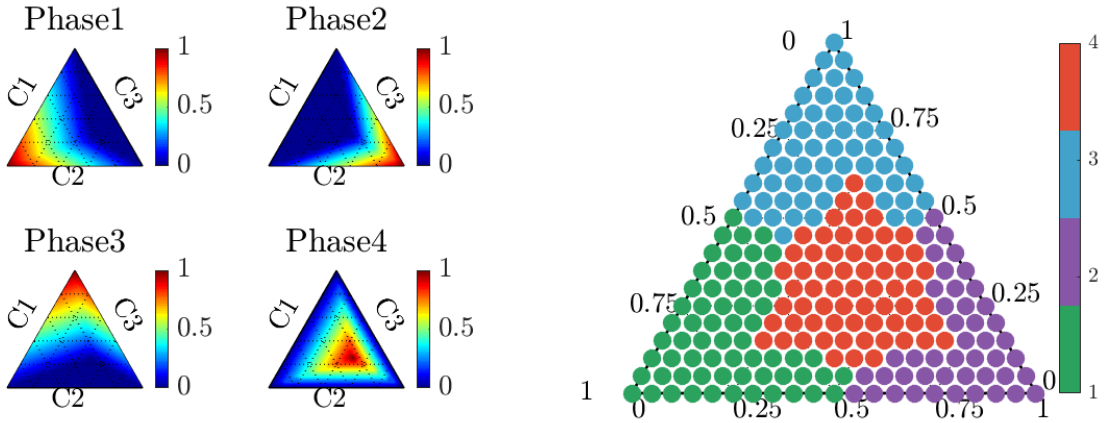


Figure S2: A contour plot of DOFs highlighting the highly overlapping behavior of each of the DOFs as opposed to Figure S1

To identify optimal number of classes, we define a loss function to be minimized. Here, we minimize the stochastic neighborhood distortion of data before and after the partition into classes. A minimum value of the loss function signifies the least distortion of the neighborhood due to partitioning. The loss function is defined as KL divergence between the two distributions calculated: Gaussian distribution of the high dimensional data and student t-distribution of the composition space and summed over all tasks. A Gaussian distribution fitted among the high dimensional responses after the partition is modified such that distance between data points of different classes in a task is set to infinity (i.e., zero probability of that being a neighbor). Gaussian and student t-distribution are computed using the method proposed for tSNE algorithm in.[3] The optimal number of classes (n) is computed using the default Bayesian optimization settings in MATLAB.

Synthetic XRD data set case study

In the main document, we performed the analysis for XRD data sets from the high-throughput experiments. Here, we report results from a similar analysis but on synthetic XRD signals. We used `synthinst133_t1_n21_p100_s1_inst` dataset from an open-source database.[4] This dataset corresponds to a ternary system with four underlying phases. The dataset contains 231 diffractograms with intensities collected at 650 q-values. Figure S3a depicts the fraction of input four phases shown as contour diagrams. Figure S3 shows a reference phase diagram, which is a result of assigning a phase to a sample in the ternary based on its largest phase content of the four underlying phases.



(a) (b)

Figure S3: (a) Phase contents shown as contour diagrams (b) Labelled space with four classes: classes are assigned based on the maximum phase content of the input phase (left panel).

To learn the metric, we follow the protocol described in the main document. In Table S1, we present the mean values from the four clustering performance measures used. Interestingly, correlation has been adjudged the best distance measure based on the mean performance measure for all the four measures. Nevertheless, mean performance measures suggest that the proposed approach (MT-LMNN) is in the top three well performing distance measure (correlation, cosine, and MT-LMNN).

Table S1: Mean value of four different performance measures for all the distance measures studied. Distance measure with best mean value for a given performance measure is highlighted (bold).

Measure	AMI	ARI	FMS	NMI
Chebychev	0.49	0.46	0.61	0.53
city block	0.45	0.42	0.59	0.49
correlation	**0.59**	**0.59**	**0.71**	**0.64**
cosine	0.55	0.53	0.66	0.60
Euclidean	0.45	0.43	0.59	0.49
Hamming	0.38	0.32	0.51	0.42
Jaccard	0.35	0.30	0.51	0.39
Minkowski	0.45	0.43	0.59	0.49
DTW	0.36	0.34	0.56	0.41
MT-LMNN	0.50	0.48	0.63	0.55
sEuclidean	0.00	0.00	0.49	0.05

Results of t-test for synthetic CV data sets

To statistically compare various distance functions, we additionally performed the statistical test with the following protocol. (1) For each of the clustering settings and distance measure, we record the predicted labels. (2) Using these labels and true labels, we compute four different clustering performance measures (ARI, AMI, NMI, FWS) from scikit-learn python library. The four performance measures: (a) the adjusted rand index (ARI)– which measures the similarity of our predicted labels with true labels ignoring permutations with a normalized chance;[5] mutual information,- which measures agreement between the true and predicted labels ignoring permutations, with two variants being used (b) adjusted mutual information (AMI), which is normalized against chance;[6] (c) normalized mutual information (NMI), which is mutual information normalized by the product of entropies of predicted and true labels;[7,8] and (d) Fowlkes-Mallows scores (FWS).[9] (3) For each distance and each performance measure, we compute the mean value of performance measures - see Table S6 for the example summary. Next, we select a distance measure with the highest mean to be the best for a given performance measure. We then perform a one-sided paired t-test between MT-LMNN and the best distance measure. We define the Null hypothesis for any

given performance measure to be: `MT-LMNN is similar to the best distance measure`. The hypothesis test described above is designed to classify how similar (or dissimilar) are two distance measures in terms of any given clustering performance measure distribution over clustering settings that are varied.

We record the hypothesis test result h obtained using the one-sided paired t-test (with a significance level of 0.01). A value of $h = 0$ signifies that the t-test failed to reject the null hypothesis while a value of $h = 1$ signifies the rejection of our null hypothesis. Table S2 summarizes the results from the t-test. If MT-LMNN is the best distance, we count number of performance measure for which it is the best. For example, MT-LMNN is the best distance according to all 4 performance measures for data set SET B1 (see Table S5). Similarly, we could number of tests for which the hypothesis was rejected and failed to be rejected.

Here, we show the results for the CV data sets considered in the main paper. Tables S3 to S6 report the mean value of four different performance measures considered for SET A1 and B2. We marked the distance measure with the highest mean using bold text. Note that, in some cases, our metric might come out to be the best distance measure for the performance measure under consideration. We also report the h-value from the paired one-sided t-test.

Table S2 displays the results for all four CV datasets. Specifically, we present the number of performance measures for which MT-LMNN came out to be best, failed to reject and reject. Please note that we focus only on cases when a measure other than MT-LMNN is the best (SET B1). In the case of SET A2, for all four performance measures the Null hypothesis failed to be rejected. As a reminder, the Null hypothesis states: MT-LMNN is similar to the best distance measure. Only for SET A2, MT-LMNN is rejected to be performing comparable to the best distance functions (i.e., Chebychev).

In summary, regardless of the performance measure used, MT-LMNN is at least as good as the best distance measure obtained from the exhaustive search. In many cases, MT-LMNN outperforms the standard similarity measure, including more sophisticated and computationally demanding distance measures such as dynamic time warping (DTW).

Table S2: The number of performance measures MT-LMNN has been classified into each of the three categories–best, failed to reject, reject–for the four synthetic data sets introduced earlier.

Data set	Best	Failed to reject	Rejected
SET A1	0	4	0
SET A2	0	0	4
SET B1	4	0	0
SET B2	0	3	1

Table S3: Mean value of four different performance measures for all the distance measures studied for SET A1 (synthetic CV data sets with separable DOFs and no noise). The best distance measures are highlighted. For this data set, MT-LMNN is failed to reject for all the performance measure as its performance is comparable to the best performing measure.

Measure	AMI	ARI	FMS	NMI
Chebychev	0.76	0.76	0.87	0.85
city block	0.74	0.75	0.87	0.83
correlation	0.24	0.28	0.62	0.29
cosine	0.33	0.35	0.65	0.38
Euclidean	**0.76**	**0.76**	0.87	**0.85**
Hamming	0.33	0.36	0.66	0.39
Jaccard	0.33	0.36	0.66	0.39
Minkowski	0.76	0.76	0.87	0.85
DTW	0.73	0.77	**0.88**	0.81
MT-LMNN	0.76	0.76	0.87	0.85
sEuclidean	0.21	0.21	0.62	0.27
h	0.00	0.00	0.00	0.00

Pseudo code and source code

We created the archive with our codes to generate the synthetic CV data sets and made them available. The archive can be downloaded from the repository on github: `https://github.com/kiranvad/MLCD`. The archive contains all MatLab codes and the set of CV curves that were used in our work. In Algorithm 1, we also include the pseudo code of the main routine.

Table S4: Mean value of four different performance measures for all the distance measure studied for SET A2(synthetic CV data sets with not-separable DOFs and no noise). For this data set, Chebychev is selected as the best distance measure for all of the performance measures used and MT-LMNN is rejected to be performing comparable to Chebychev.

Measure	AMI	ARI	FMS	NMI
Chebychev	**0.75**	**0.74**	**0.85**	**0.84**
city block	0.56	0.54	0.72	0.63
correlation	0.29	0.30	0.59	0.33
cosine	0.28	0.30	0.59	0.33
Euclidean	0.72	0.72	0.84	0.81
Hamming	0.25	0.25	0.58	0.28
Jaccard	0.25	0.25	0.58	0.28
Minkowski	0.72	0.72	0.84	0.81
DTW	0.54	0.54	0.74	0.62
MT-LMNN	0.54	0.53	0.71	0.61
sEuclidean	0.30	0.31	0.60	0.36
h	1.00	1.00	1.00	1.00

Table S5: Mean value of four different performance measures for all the distance measure studied for SET B1(synthetic CV data sets with not-separable DOFs with noise). For this test case MT-LMNN is adjudged the best distance measure for all of the performance measures considered.

Measure	AMI	ARI	FMS	NMI
Chebychev	0.36	0.36	0.64	0.42
city block	0.32	0.34	0.65	0.38
correlation	0.17	0.16	0.58	0.24
cosine	0.23	0.25	0.62	0.28
Euclidean	0.41	0.40	0.66	0.47
Hamming	0.25	0.27	0.60	0.28
Jaccard	0.25	0.27	0.60	0.28
Minkowski	0.41	0.40	0.66	0.47
DTW	0.38	0.38	0.64	0.43
MT-LMNN	**0.41**	**0.44**	**0.68**	**0.47**
sEuclidean	0.22	0.21	0.62	0.28

Table S6: Mean value of four different performance measures for all the distance measure studied for SET B2 (synthetic CV data sets with Not-separable DOFs with a Gaussian noise). Note MT-LMNN performance is comparable to more sophisticated DTW.

Measure	AMI	ARI	FMS	NMI
Chebychev	0.27	0.24	0.54	0.31
city block	0.25	0.24	0.55	0.29
correlation	0.26	0.26	0.57	0.30
cosine	0.27	0.27	0.58	0.31
Euclidean	0.27	0.24	0.54	0.31
Hamming	0.25	0.25	0.58	0.28
Jaccard	0.25	0.25	0.58	0.28
Minkowski	0.27	0.24	0.54	0.31
DTW	**0.32**	**0.32**	0.60	**0.37**
MT-LMNN	0.25	0.22	0.60	0.31
sEuclidean	0.29	0.30	**0.60**	0.34
h	0.00	1.00	0.00	0.00

Algorithm 1: MLCD

Data: Response Matrix X, Composition Matrix C
Result: Metric M_0

Initialization;
for $t = 1$ *to* 3 **do**
    y_t = definetasks(C);
end
▷ `Run a Bayesian Optimization to find` γ`'s`
while *Budget remains* **do**
    ▷ `Compute` M_0 `using MT-LMNN`
    M_0 = mtlmnn(X,y_t);
    ▷ `Define function to be optimized as k-NN Classification error (knncl)`
    error = knncl(M_0,X,y_t);
    ▷ `Perform Bayesian optimization until Budget expires`
    bayesopt(error,domain);
end

References

(1) Snoek, J.; Larochelle, H.; Adams, R. P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems. 2012; pp 2951–2959.

(2) Wu, J.; Toscano-Palmerin, S.; Frazier, P. I.; Wilson, A. G. Practical multi-fidelity bayesian optimization for hyperparameter tuning. *arXiv preprint arXiv:1903.04703* **2019**,

(3) Maaten, L. v. d.; Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **2008**, *9*, 2579–2605.

(4) Bernstein, R. L. B. R.; Suram, J. M. G. S. K.; Selman, C. P. G. B.; van Dover, R. B. A computational challenge problem in materials discovery: Synthetic problem generator and real-world datasets. **2014**,

(5) Santos, J. M.; Embrechts, M. On the use of the adjusted rand index as a metric for evaluating supervised classification. International conference on artificial neural networks. 2009; pp 175–184.

(6) Hubert, L.; Arabie, P. Comparing partitions. *Journal of classification* **1985**, *2*, 193–218.

(7) Strehl, A.; Ghosh, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* **2002**, *3*, 583–617.

(8) Vinh, N. X.; Epps, J.; Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* **2010**, *11*, 2837–2854.

(9) Fowlkes, E. B.; Mallows, C. L. A method for comparing two hierarchical clusterings. *Journal of the American statistical association* **1983**, *78*, 553–569.