[Supplementary] RD-DPP: Rate-Distortion Theory Meets Determinantal Point Process to Diversify Learning Data Samples

A. Proof of Theorem 1

Proof. To prove Theorem 1, we need to show that $\frac{1}{2}\log\det\left(\mathbf{I}+\alpha\mathbf{Z}\mathbf{Z}^\top\right) = \frac{1}{2}\log\det\left(\mathbf{I}+\alpha\mathbf{Z}^\top\mathbf{Z}\right)$ (b), because equality (a) is the definition of RD, and (c) is proven to be the normalization term of DPP. Recall the data matrix is $\mathbf{Z}\in\mathbb{R}^{d\times n}$. The Singular Value Decomposition (SVD) of $\mathbf{Z}$ is

$$\mathbf{Z} = \mathbf{U}_{d\times d}\mathbf{S}_{d\times n}\mathbf{V}_{n\times n}^\top, \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary, and the diagonal elements of $\mathbf{S}$ are the singular values of $\mathbf{Z}$. Suppose the rank of $\mathbf{Z}$ is r meaning that the first r diagonal elements of S are greater than 0. The rest of the diagonal elements and all off-diagonal elements are zero. Hence,

$$\mathbf{Z}\mathbf{Z}^\top = \mathbf{U}\mathbf{S}\mathbf{V}^\top\mathbf{V}\mathbf{S}^\top\mathbf{U}^T = \mathbf{U}(\mathbf{S}\mathbf{S}^\top)\mathbf{U}^T, \tag{2}$$

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{V}\mathbf{S}\mathbf{U}^\top\mathbf{U}\mathbf{S}^\top\mathbf{V}^T = \mathbf{V}(\mathbf{S}^\top\mathbf{S})\mathbf{V}^T. \tag{3}$$

Since $\mathbf{U}(\mathbf{S}\mathbf{S}^\top)\mathbf{U}^T$ and $\mathbf{V}(\mathbf{S}^\top\mathbf{S})\mathbf{V}^T$ are the SVD decomposition of $\mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{Z}^\top\mathbf{Z}$, they have the same non-zero singular values. Also, $\mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{Z}\mathbf{Z}^\top$ are *positive semi-definite* (PSD), which implies that their eigenvalues and singular values are the same. Therefore,

$$\log\det\left(\mathbf{I}+\alpha\mathbf{Z}\mathbf{Z}^\top\right) = \sum_{i=1}^{r}\log(\alpha\lambda_i+1) + \sum_{i=1}^{d-r}\log(1) \tag{4}$$

$$= \sum_{i=1}^{r}\log(\alpha\lambda_i+1),$$

$$\log\det\left(\mathbf{I}+\alpha\mathbf{Z}^\top\mathbf{Z}\right) = \sum_{i=1}^{r}\log(\alpha\lambda_i+1) + \sum_{i=1}^{n-r}\log(1) \tag{5}$$

$$= \sum_{i=1}^{r}\log(\alpha\lambda_i+1),$$

where λ_i is the ith singular value of $\mathbf{Z}\mathbf{Z}^\top$ or $\mathbf{Z}^\top\mathbf{Z}$. Therefore, Eqs. 2 and 4 are equal. □

B. Packet Preparation and Experiment Setup

B.1. Packet Preparation

In practical systems, the data is transmitted in the form of packets, which may result in an extreme non-i.i.d scenario. In our experiment, we assume each packet contains the same number of samples. We also assume data from the same packets are very similar, and we do not operate the intra-packet (i.e. any operations in a packet, such as permutation of the order).

To generate packets with this assumption, we first use the entire training set to train a random neural network. Then, use their representation from this trained network to perform K-means clustering to generate 100 clusters. Noting that generally, the number of samples in each cluster is not the same. Thus, in each experiment, we sample the same number of samples from each cluster (e.g., 64 for MNIST and FMNIST) to encapsulate into a packet, and naturally, we have a total of 100 packets for transmission. We do a similar preparation for CIFAR10 (a total of 100 packets and each packet contains 200 samples) and UCI datasets (a total of 60 packets and each packet contains 5 samples). The composition of some exemplary packets is shown in Fig. 1.

We define the feature of each packet as follows, which is in a relatively fine-grained way and invariant to the order of samples,

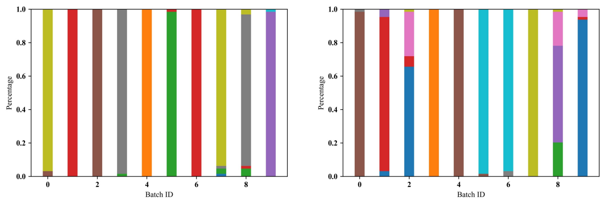
$$f(\mathbf{X}_i) := \Big[\frac{1}{|C_1^i|}\sum_{e\in C_1^i}\mathbf{z}_e, \frac{1}{|C_2^i|}\sum_{e\in C_2^i}\mathbf{z}_e, \tag{6}$$

$$\dots, \frac{1}{|C_{c_T}^i|}\sum_{e\in C_i^{c_T}}\mathbf{z}_e\Big] \in \mathbb{R}^{dc_T},$$

where C_j^i denotes the index set of class j in packet $\mathbf{X}_i$, and $\frac{1}{|C_j^i|}\sum_{e\in C_j^i}\mathbf{z}_e \in \mathbb{R}^d$ denotes the averaged features of class j. We normalize the obtained vector to have $\|f(\mathbf{X}_i)\| = 1$.

B.2. Training Detail

We set $\varepsilon^2 = 0.5$ in Eqs. (16) for all experiments.



(a) MNIST **(b) FMNIST**

Figure 1. The visualization of cluster-based data splitting. Each color denotes a class.

B.2.1 MNIST and FMNIST

We set $K = 5, \phi_0 = 2$. The network architecture for MNIST and FMNIST is presented in Table 1.

Table 1. The architectural details of the network used in MNIST and FMNIST.
It outputs the similarity of the input and the target.

Layer Type	**Kernel Size** $K_1 \times K_2 \times C_{in} \times C_{out}$
Conv2d+ReLU	$3 \times 3 \times 1 \times 8$
MaxPool2d	-
Conv2d+ReLU	$3 \times 3 \times 8 \times 16$
MaxPool2d	-
Conv2d+BatchNorm+ReLU	$3 \times 3 \times 16 \times 32$
MaxPool2d	-
Full-connected	$1 \times 1 \times 288 \times 10$

All models use an ADAM optimizer with a learning rate of 1e-3 and a mini-batch size of 64. We train each model with 100 epochs and report the average of the last 20 epochs' test accuracy as the final accuracy of the model. The other experiments use the same way to report.

B.2.2 CIFAR10

The models we are using can be found in https://github.com/kuangliu/pytorch-cifar.

We set $K = 10$. We use SGD optimizer in this experiment. We set the learning rate to 0.01, the momentum factor to 0.9, and the weight decay factor to 5e-4. We also use a cosine annealing schedule and set T_{max} to 200.

B.2.3 UCI Datasets

Here, we set the total number of packets to 60 and each packet has 5 samples. The selection is initialized with 3 packets, and in each round, we select $k = 3$ packets by different approaches.

All datasets are split into 70%-30% training and test subsets and pre-processed by Z-score normalization. We set $K = 3$. The learning rate was set to 1e-2, and 1e-3 for Yeast, Cardiotocography, and Statlog, respectively.

B.2.4 Linear Evaluation Protocol

All models use an ADAM optimizer with a learning rate of 1e-3 and a mini-batch size of 64.

B.3. Datasets Setup of Experiment in Discussion

Rotated MNIST In each sub-task, the digits were rotated by a pre-defined angle. Each task in Rotation MNIST is a 10-class classification problem where their labels are the corresponding digits. Thus, each subsequent task involves classification on the same ten digits.

MNIST Fellowship The MNIST Fellowship is a combination of MNIST, Fashion MNIST, and KMNIST. Each sub-task corresponds to one dataset with ten classes of annotation. This task has more various domain differences than the previous one.

C. Additional results

C.1. Experiment on Raw Samples

In the above experiments, we showed the utility of our dual-mode selection method when applied to the representation of data samples in lower-dimensional spaces, like the feature vectors exploited from the latest layers of deep learning architectures. We can apply our method to raw samples as well. In this respect, we evaluate our method using three UCI small datasets: Yeast [1], Cardiotocography [2], and Statlog (Landsat Satellite) [3]. Here, we set the total number of packets to 60 and each packet has 5 samples. The selection is initialized with 3 packets, and in each round, we select $k = 3$ packets by different approaches. Since their samples are limited, we only use logistic regression as the learning model here. To encounter the unbalanced data, we use *Area Under the Receiver Operating Characteristic Curve* (AUCROC) to assess their performances and present the results (average of 10 runs) in Table 2. Again, our proposed selection method outperforms the random selection on Yeast, Cardiotocography, and Statlog datasets with 3%-7%, 1%-4%, and 0.5%-4% gain at transmission budgets 3, 9, and 15, respectively.

Table 2. Performance (AUCROC) on three UCI datasets.

Dataset	**Yeast**			**Cardio.**			**Statlog.**		
Budget	**3**	**9**	**15**	**3**	**9**	**15**	**3**	**9**	**15**
RD-DPP	**75.16**	**81.53**	**84.33**	**74.08**	**85.07**	**91.47**	**88.92**	**94.22**	95.16
Uncertainty Dec.	73.35	79.81	83.11	67.41	84.59	90.85	82.26	93.68	**95.26**
Min Margin Dec.	71.29	75.97	81.36	67.57	75.9	83.42	84.13	86.48	93.9
Rand	68.15	77.62	79.78	72.51	81.9	86.61	84.42	89.89	94.64

References

[1] Yeast. UCI Machine Learning Repository, 1996. DOI: 10.24432/C5KG68. 2

[2] J. Campos, D. & Bernardes. Cardiotocography. UCI Machine Learning Repository, 2010. DOI: 10.24432/C51S4N. 2

[3] Ashwin Srinivasan. Statlog (Landsat Satellite). UCI Machine Learning Repository, 1993. DOI: 10.24432/C55887. 2