CS771 - Intro to ML (Autumn 2024): Mini-project 2

Due Date: November 22, 2024 (11:59pm)

1 Problem 1

You are provided 20 training datasets $D_1, D_2, \ldots, D_{20}$, each of which is a subset of the CIFAR-10 image classification dataset. Out of these 20 datasets, the first 10 datasets $D_1, D_2, \ldots, D_{10}$ have their inputs from the same distribution (which means the input features' characteristics of datasets D_1 to D_{10} are similar to each other, or equivalently we can say that $p(x)$ is the same for all these datasets), whereas $D_{11}, D_{12}, \ldots, D_{20}$ have their inputs from 10 different input distributions (thus $p(x)$ is different for each of these datasets). However, input distributions of $D_{11}, \ldots, D_{20}$ have some degree of similarity with the common input distribution that the inputs in D_1 to D_{10} come from.

Datasets D_1 to D_{10} and datasets D_{11} to D_{12} are provided in two separate folders (download link is provided below at the end of this section). The inputs are given in form of the raw images (32×32 size color images) and you are free to use any suitable feature representation for the inputs (e.g., use a pre-trained neural net to extract features, or you may even use some kernel if you want!).

Also note that the first dataset D_1 is labeled and the remaining 19 datasets are unlabeled. However, we also provide 20 held-out labeled datasets $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{20}$ which you can use to assess the performance (accuracy) of the models you will train. Note that you must NOT use the heldout datasets for anything else (e.g., cross-validation) but only for evaluating the model accuracy.

1.1 Task 1

Using D_1, train a model f_1 using an LwP classifier (you are free to use any feature representation or kernels, and are also allowed to use your own improved variant of the basic LwP classifier if it helps solve the task better), and then apply f_1 to predict the labels of D_2 (note that its true labels are not known), use D_2 along with its predicted labels to update f_1 to f_2. How you use the predicted labels of D_2 to update f_1 to f_2 is up to you but, when doing this, you must not use the dataset D_1 (note that you are using f_1 anyway which was learned using D_1).

Now use f_2 to predict the labels of D_3 and update f_2 to f_3 in the same way as you updated f_1 to f_2. You have to keep doing this for the first 10 datasets $D_1, D_2, \ldots, D_{10}$ and learn 10 models $f_1, f_2, \ldots, f_{10}$. Note that the size of each model (in terms of the number of parameters) should be the same.

Once you have learned them, apply each model $f_i \in \{f_1, f_2, \ldots, f_{10}\}$ on the i^{th} heldout dataset $\hat{D}_i$ as well as the previous heldout datasets $\hat{D}_j, j < i$ (e.g., apply f_3 on $\hat{D}_3$ and also on $\hat{D}_1, \hat{D}_2$), and report the accuracies of each of these models on each of these heldout datasets. In your report, show the various accuracies in form of a matrix (with 10 rows representing the 10 models and 10 columns representing the 10 held-out datasets).

Apart from achieving good accuracies, your goal should also be to ensure that when a model is updated, the performance on previous datasets should not degrade significantly. For example, if you are updating f_6 to f_7, then not only should the model f_7 give good accuracy on $\hat{D}_7$ but its performance on the previous heldout datasets $\hat{D}_1$ to $\hat{D}_6$ should ideally not be worse as compared to the performance of the previous model f_6 on these heldout datasets.

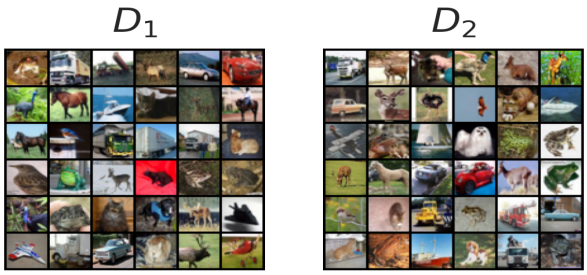

Figure 1: Sample images from datasets: D_1 and D_2. Recall the problem statement (refer 1.1); each of the dataset originates from the same input distribution $p(x)$.

1.2 Task 2

Next, starting with f_{10} from task 1 as the initial model, repeat the same procedure using the datasets $D_{11}, D_{12}, \ldots, D_{20}$ (again, one dataset at a time) to learn models $f_{11}, f_{12}, \ldots, f_{20}$. Just like task 1, how you update the models $f_{11}, f_{12}, \ldots,$ from one to the next one is up to you but you should consider the fact that, unlike task 1 where the datasets $D_1, D_2, \ldots, D_{10}$ had the same input distribution, the datasets $D_{11}, D_{12}, \ldots, D_{20}$ are possibly from distributions that are slightly different from each other (see Figures below for an illustration, or look at the dataset images in the dataset folder). You may want to consider this aspect when you update the models (the model update method you used for task 1 may not be good enough for task 2).

Just like you did for task 1, once you have learned $f_{11}, f_{12}, \ldots, f_{20}$, apply each model $f_i \in \{f_{11}, f_{12}, \ldots, f_{20}\}$ on the i^{th} heldout dataset $\hat{D}_i$ as well as the previous heldout datasets $\hat{D}_j, j < i$ (e.g., apply f_{13} on $\hat{D}_{13}$ and also on $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{12}$), and report the accuracies of

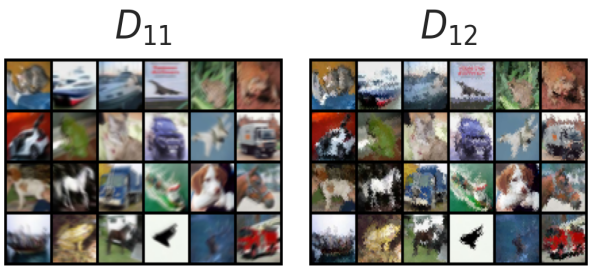

Figure 2: Sample images from datasets: D_{11} and D_{12}. Recall the problem statement (refer 1.2); each of the dataset originates has a different input distribution $p(x)$.

each of these models on each of these heldout datasets. In your report, show the various accuracies in form of a matrix (with 10 rows representing the 10 models and 20 columns representing the 20 held-out datasets).

Just like task 1, apart from achieving good accuracy, your goal should be to ensure that when a model is updated, the performance on previous datasets should not degrade significantly. For example, if you are updating from f_{16} to f_{17}, then not only should the model f_{17} give good accuracy on $\hat{D}_{17}$ but its performance on the previous heldout datasets $\hat{D}_1$ to $\hat{D}_{16}$ should ideally not be worse as compared to the performance of the previous model f_{16} on these heldout datasets.

1.3 Deliverables for Task 1 and Task 2, and Dataset

In a single ZIP file, using the same naming convention as mini-project 1, you should submit the following:

- Your code in form of Python notebooks, one for task 1 and one for task 2. Please add comments, where necessary, in your code to help explain what the code is doing.
- A description of your approach in about 2-3 pages using the same LaTeX template that you used for mini-project 1, focusing mainly on the final approach that you eventually took.

The dataset for task 1 and task 2 can be downloaded from this URL:
`https://tinyurl.com/cs771-mp2-data`

Please do not submit the dataset with your ZIP file submission.

Here is the submission URL for your ZIP file:
`https://www.dropbox.com/request/wYH5x6dAhEtI4yjv8e43`

2 Problem 2

For this problem, your task will be to read the following two papers (these are on the topic of continual/lifelong unsupervised domain adaptation) and pick ONE of these papers to make a video presentation (slides with your voice-over) of approximately 5 minutes duration, summarizing the problem studied in the paper, the key ideas proposed, and the results presented in the paper:

- Lifelong Domain Adaptation via Consolidated Internal Distribution (NeurIPS 2021)
- Deja vu: Continual model generalization for unseen domains(ICLR 2023)

Upload your presentation's video on a YouTube channel and share the link in the project report that you created for Problem 1.

Note: Reading these papers, you would also realize that the problem settings studied in these papers are similar to the setting given in Problem 1 of this mini-project. You are free to use the understanding/insights gained from reading these papers (or any other related research papers on this topic that you may find on your own) to solve Problem 1. Note however that your approach should still be using LwP or some variant of LwP, and not some advanced model such as a deep neural network.