HW 3: Deep Ensembles & Covariate Shift

Version: May 11, 2023

This homework will deal with nonlinear classification under *covariate shift* – the test set distribution is different from the training set distribution. We will use ensembling to improve test performance and confidence calibration.

1. Download the train, validation, and test datasets (using code template)
2. Construct a classifier and train it.
    a. Define a Pytorch module with a simple multi-layer perceptron architecture
    b. Each classifier is randomly initialized and trained on the data
    c. Repeat 10 times to make an ensemble of classifiers
3. Analyze and visualize the results. Detailed instructions about what should be shown are given in the write-up section.
    a. The dataset features are 2-d so you can visualize the classifiers and data
    b. Accuracy and expected calibration error, for train, validation, and test, comparing individual models to the ensemble model.
4. Submit code and write-up on canvas.

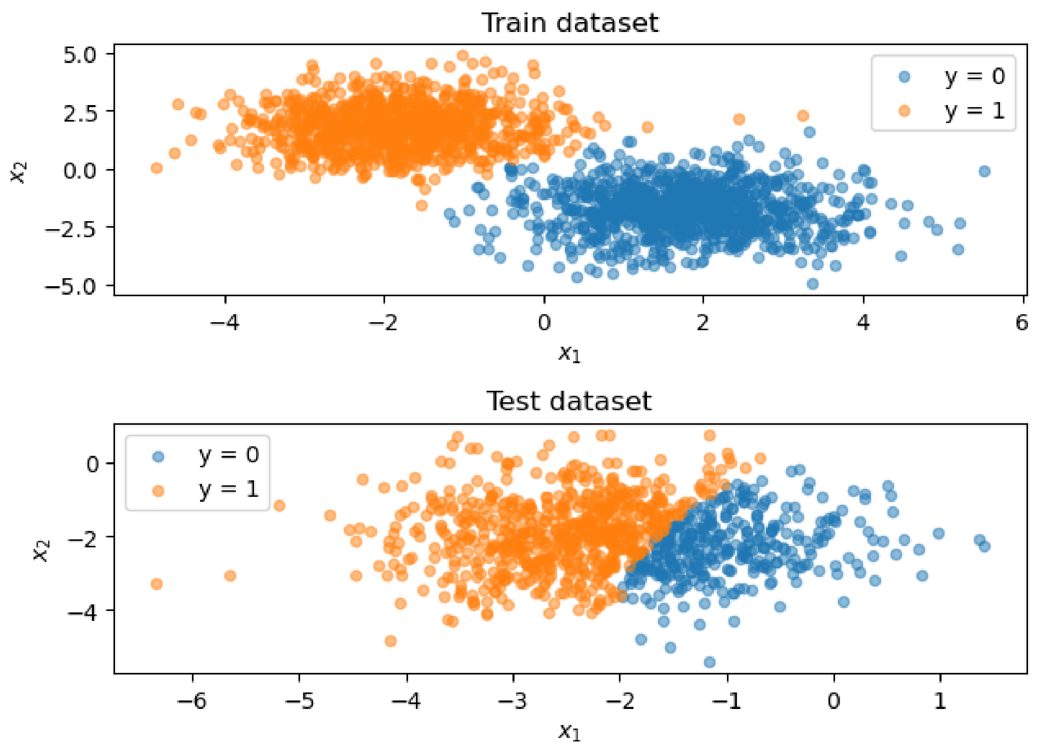

Fig.1 A picture of the nonlinear classification on the training dataset.

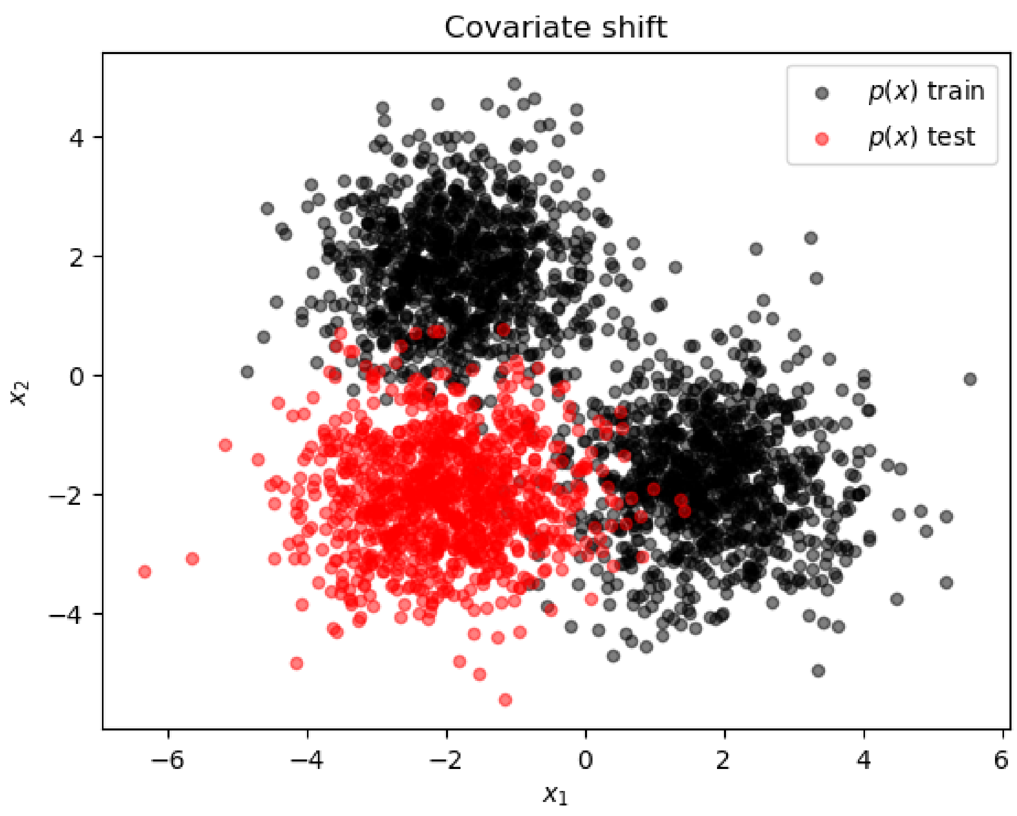

Fig. 2 A picture of the covariate shift between train and test data. p(x) changes, but the labeling function, p(y|x) does not change.

Extra credit

Try different approaches to improve test accuracy and calibration.

- Mixup might work well here
- Bayesian sampling with SGLD, instead of using SGD
- Compare Stochastic Weight Averaging, or SWAG
- Try something from domain generalization methods leaderboard: wilds.stanford.edu
- Try MC-Dropout - i.e. train with dropout and do dropout at inference time, ensembling predictions with different dropout noise.

Code:

Template code is included, with TODOs of things to fill in. Comments also contain info about points and requirements. Please keep the function names for easier grading. (If you want to try a complex strategy that requires refactoring, just put a comment in the predefined function telling us where to look.) We prefer Ipython notebooks for the code (also easier if you're running on Google Colab).

Write-up instructions:

The write-up should be as succinct as possible. Include the following sections, with the requirements in each section. (And it should not be a PDF of the Ipython notebook.)

(a) **Describe architecture and hyper-parameter choices (2 points)** Report any hyper-parameter choices you made. For instance, learning rate and optimizer hyper-parameters, number of epochs of training, any preprocessing steps. For the architecture, the number of layers and units in each layer and nonlinearity used, and whether you used normalization layers or other regularizers.

(b) **Visualize** Show a scatter plot of validation and test data, with all the classifier boundaries (each of ten models and the ensemble model) superimposed on top. All of this should appear in a single plot **(3 points).**

(c) **Calculate** accuracy and expected calibration error for each model for both train/validation/and test data. Show the mean values across the 10 models, and the performance of the (single) ensemble model. Give a table of results like this. **(3 points)**

	Mean across models	Ensemble model
Train accuracy		
Validation accuracy		
Test accuracy		
Train ECE		
Validation ECE		
Test ECE		

(d) **Discuss (2 points)** Discuss how ensembling affects results under covariate shift in the test domain, referencing your table of results and visualization.

(e) Extra credit (optional)

Collaboration:

Feel free to discuss ideas with classmates, but you should do the coding and write-up on your own. AI tools are allowed.