CS464 Introduction to Machine Learning
Spring 2020
Homework 2

Due: December 11, 2020 11:59 PM

Instructions

- For this homework, you may use any programming language of your choice.
- You are NOT allowed to use any machine learning packages, libraries or toolboxes for this assignment (such as scikit-learn, tensorflow, keras, theano, MATLAB Statistics and Machine Learning Toolbox functions, e1071, nnet, kernlab etc.) unless otherwise stated.
- Submit a soft copy of your homework to Moodle.
- Upload your code and written answers to the related assignment section on Moodle (.TAR or .ZIP). Submitting hard copy, handwritten or scanned files is NOT allowed.
- The name of your compressed folder must be "`CS464_HW2_Section#_Firstname_Lastname`" (i.e., `CS464_HW2_1_john_doe`). Please do not use any Turkish characters in your compressed folder name.
- Your code should be in a format that is easy to run and must include a driver script serving as an entry point. You must also provide a README file with clear instructions on how to execute your program.
- This is an individual assignment for each student. That is, you are NOT allowed to share your work with your classmates.
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.
- Plots generated for this homework should be properly formatted. That is include title, axis labels and legend (if needed) in your plots.
- **You may ask your questions regarding this homework to your TA, Doruk Çakmakçı (doruk.cakmakci@bilkent.edu.tr)**

1 PCA on Van Gogh Paintings [20 pts]

In this question, you are expected to analyze paintings of Van Gogh using PCA. You will use the dataset provided within the homework zip file as `q1_dataset.zip`. The dataset consists of 877 paintings. For this question, use of PCA functions from any library is **not allowed**. However, you may use functions to perform singular value decomposition (SVD). Apply the preprocessing mentioned below to the dataset.

Preprocessing: First, if an image is grayscale, stack the 64×64 image along the third dimension to obtain RGB version of size $64 \times 64 \times 3$. Flatten all images of size $64 \times 64 \times 3$ to get 4096×3 matrix for each image. Note that all images are 3-channel RGB. Create a 3-D array, X, of size $D \times 4096 \times 3$ by stacking flattened matrices of the images provided in the dataset. Slice X as $X_i = X[:, :, i]$ where i corresponds to the first three index, for obtaining each color channel matrix independently for all images. Reshape all X_i to obtain

matrices, instead of 3D arrays.

Question 1.1 **[10 pts]** Apply PCA on color channel matrices (i.e. X_i) using SVD based approach to obtain first 100 principal components per color channel matrix for each dataset. Sort the first 100 singular values in descending order and plot a bar chart per X_i. Report proportion of variance explained (PVE) by the first 10 principal components. Discuss your results.

Question 1.2 **[10 pts]** For this part, you will work on a noisy dataset generated from Van Gogh paintings. First, compute the mean (μ_D) and variance (σ_D^2) of the dataset. You should have a matrix of size $64 \times 64 \times 3$ per mean and variance. Further, add noise to each image by sampling from a $64 \times 64 \times 3$ dimensional gaussian with mean μ_D and variance σ_D^2. Scale the noise with a factor of 0.01 before they are added to the images. Compute the color channel matrices for the noisy dataset using the preprocessing procedure mentioned previously. Apply PCA using SVD based approach to obtain the first 100 principal components on each color channel matrix. Sort the first 100 singular values corresponding to the principal components and plot a bar chart. Discuss the effect of artificially added noise on the principal components and corresponding singular values based on your results. Describe how you can reconstruct images from principal components. How does the number of principal components used for reconstruction, affects the reconstructed images. Discuss how you can decrease the presence of artificially added noise on the images reconstructed from principal components.

2 Linear Regression on University Admission Records [30 pts]

For this question, you will work with `q2_dataset.csv`, which is a dataset of graduate admission records of 500 students. Here, you can not use any machine learning library. You will train regression models on the dataset to predict the admission chance of a student to a university. Chance of admission will be the response variable and remaining features will be the predictor variables. Features consist of the following: GRE Score, TOEFL Score, University rating, Statement of Purpose, Letter of Recommendation, CGPA and Research. Research is a binary variable indicating whether the student conducted any research activity prior to application and remaining features are continuous variables.

Question 2.1 **[4 pts]** Derive the general closed form solution for multivariate regression model using ordinary least squares loss function given in Eqn. 2.1 Briefly explain each matrix involved in calculation and how they are constructed.

$$J_n = ||y - X\beta||^2 = (y - X\beta)^T(y - X\beta) \tag{2.1}$$

Question 2.2 **[5 pts]** For this question, you will train linear regression models, using the formula you have derived for Question 2.1, to predict the chance of admission based on the remaining statistics. Using any machine learning packages, libraries or toolboxes is not allowed. First, shuffle and split the dataset to 5 folds where each fold contains 100 samples. Train linear regression models using the following setup: separate fold i as test set and train a model on the remaining folds. In the end, you should have trained 5 models, one per test fold. For each model, calculate R^2, mean squared error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). MAPE for a dataset of N samples should be calculated using Eqn. 2.2 where A_t and P_t are ground truth and predicted chance of admission, respectively. **Record the results calculated on the test folds** and report your results.

$$M = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_t - P_t}{A_t}\right| \tag{2.2}$$

Question 2.3 **[5 pts]** For this question, you will train linear regression models with L_1 regularization (i.e. Lasso) in order to predict chance of admission. Use the experimental setup described in Question 2.2 (i.e.

calculate the mentioned performance metrics on a 5-fold cross validation setup). Assume that regularization penalty factor (i.e. λ) is 1. **Record the results calculated on the test folds** and report your results.

Question 2.4 [16 pts] In this part, you will compare the performance of the models trained for Question 2.2 and 2.3 with respect to the calculated performance metrics. First plot a boxplot with results of both models, per performance metric. That is, you should plot 4 boxplots each having two boxes. Compare the performance of models. Then, discuss the following items in detail:

- Discuss the behavior of MSE and MAE with respect to their response to outliers in the dataset. Also, visually compare the performance of models with respect to MAE and MSE boxplots. Why performance of the models are more similar when they are compared in terms of MSE?
- Suppose you are going to select a single performance metric among the considered options. which would you select? Discuss.
- Discuss why a cross validation based experimental setup is prefered, opposed to reporting results on a fixed test dataset. (Hint: build your discussion on dataset size).
- The effect of lasso regularization on model weights. Discuss why L_1 regularization might not be beneficial on the model performance. (Hint: Think about correlation among features).
- Assume that the dataset used in this question was of size 50000. In this case, would you prefer a cross validation based approach or a fixed test dataset approach. State your reasoning.

3 Logistic Regression for Survival Prediction [25 pts]

For this question, you will predict passenger survival in the Titanic disaster using a logistic regression model. Here, you will work on the training and test datasets provided in homework zip file as `q3_train_dataset.csv` and `q3_test_dataset.csv`, respectively. You should analyse the dataset and normalize features if needed. For this question, use of any machine learning packages, libraries or toolboxes is not allowed. Dataset contains the following fields:

- **Survival Status**: A binary variable which indicates if the passenger survived the disaster.
- **Ticket Class**: A categorical variable which indicates ticket type (i.e. 1,2 or 3) purchased by the passenger.
- **Gender**: Gender of the passenger (i.e. "male" or "female").
- **Age**: Age of the passenger.
- **Siblings / Spouse**: The number of sibling or spouses of the passenger aboard the Titanic.
- **Parent / Children**: The number of parents or children of the passenger aboard the Titanic.
- **Fare**: Passenger fare.
- **Port of Embarkation**: Cherbourg (C), Queenstown (Q) or Southampton (S).

You should analyse the dataset and normalize the features if needed. Also, measure and report training times of models.

Question 3.1 [12 pts] Implement mini-batch gradient ascent algorithm with *batch size = 32* and stochastic gradient ascent algorithm to train logistic regression models. Initialize all weights to random numbers drawn from a Gaussian distribution $N(0, 0.01)$. Choose the learning rate from the following set: $\{10^{-4}, 10^{-3}, 10^{-2}\}$. Perform 1000 iterations to train your model. Report class based accuracies, accuracy, precision, recall, negative predictive value (NPV), false positive rate (FPR), false discovery rate (FDR), F1, F2 scores and

the confusion matrix using your model on the corresponding test set. For your calculations, assume positive class is passenger survived. Discuss your results.

Question 3.2 [**8 pts**] Implement full batch gradient ascent algorithm to train your logistic regression model. Initialize all weights to random numbers drawn from a Gaussian distribution $N(0, 0.01)$. Choose the learning rate from the following set: $\{10^{-4}, 10^{-3}, 10^{-2}\}$. Perform 1000 iterations to train your models. Print model weights in each iteration $i \in \{100, 200, ..., 1000\}$. Report class based accuracies, accuracy, precision, recall, negative predictive value (NPV), false positive rate (FPR), false discovery rate (FDR), F1, F2 scores and the confusion matrix using your model on the given test set. For your calculations, assume positive class is passenger survived. Discuss your results.

Question 3.3 [**5 pts**] Discuss how you can infer feature importance from logistic regression model weights. Discuss why feature normalization is needed for comparing feature importance based on logistic regression model weights. Is it possible to compare importance of categorical and continuous features together? If it is possible, report the two most important features for the model trained in Question 3.2. If it is not possible, report the most important categorical feature and continuous feature for the model trained in Question 3.2. Also, discuss the effect of batch size on the training time of a logistic regression model.

4 SVM [25 pts]

In this question, you are expected to classify flowers images using multi-class SVM models with one-vs-all approach. The dataset consists images from 5 classes: Daisy (Class 0), Dandelion (Class 1), Rose (Class 2), Sunflower (Class 3) and Tulip (Class 4). You will use two sets of features: (i) Raw image pixels; and (ii) Features extracted from images by a deep convolutional neural network. The features are already extracted and prepared for your use in `q4_dataset.mat`.

Specifically, we used Inception v3 model [1] as our pretrained deep convolutional neural network which was trained on ImageNet dataset [2] for classification task. Pre-training a network on one task, and using them to extract features for other image related tasks is an example of transfer learning which is widely used when the data for the task at hand is small or when you want to start from a good initial point.

For this question you can use any machine learning library (i.e. scikit-learn). The dataset is provided in homework zip file as `q4_dataset.mat` and consists of the following:

- **inception_features**: A matrix of size 1250×2048 whose rows correspond to features extracted by Inception v3.
- **images**: A tensor of size $1250 \times 64 \times 64 \times 3$ which contains 1250 RGB images of size 64×64
- **class_labels**: A matrix of size 1250×1 whose rows contains class labels

Experimental Setup

You will train each model using a stratified 5-fold cross validation (i.e. distributions of the folds are similar to whole dataset). During iteration i of setup, fold i will be the test set, fold $(i+1)$ mod 5 will be the validation set and the model will be trained on the remaining folds. Before training, select a suitable performance metric.. You should apply the following procedure in order to select best performing hyper-parameter setting based on validation set performance.

Procedure: First perform a grid search on the hyper-parameter space and select the best performing parameter setting based on validation set performance. That is, you should train models with different hyper-parameter settings on the training set and measure their performance on the validation set. Based on the validation set performance you will select the best performing hyper-parameter setting for the remaining steps. Then, you will train another model with the selected setting on training + validation sets (i.e. $3 + 1 = 4$ folds will be used for training). You are required to test your model on the test set using the

selected performance metric and record the results for further use.

In the end, you should have 5 results with respect to the selected metric, one result per test fold. Don't forget to record results, you will plot a boxplot in order to compare methods visually. Also measure and report running time of the setup for each question.

Question 4.1 [10 pts] Train a SVM model with linear kernel on features extracted by Inception v3. Tune C hyper-parameter of the model using the scheme presented in Experimental Setup. Here you should perform parameter selection from the following space: $C \in \{10^{-6}, 10^{-4}, 10^{-2}, 1, 10^{1}, 10^{10}\}$. Determine if hard margin or soft margin setting performs better.

Question 4.2 [6 pts] Train a non-linear SVM with radial basis function (RBF) kernel on features extracted by Inception v3. Radial basis function is defined as:

$$K(x, x') = exp(-\gamma||x - x'||^2) \tag{4.1}$$

where γ is an hyper-parameter of the model which determines the radius of influence of support vectors on decision boundary of the model. Tune C and γ parameters of the model using the scheme presented in Experimental Setup. Here you should perform parameter selection from the space given in Eqn. 4.2. Report the performance obtained on each test fold.

$$\{\gamma, C \mid C \in \{10^{-4}, 10^{-2}, 1, 10^{1}, 10^{10}\}, \gamma \in \{2^{-4}, 2^{-2}, 1, 2^{2}, 2^{10}, "scale"\}\} \tag{4.2}$$

where "scale" is defined as for a feature matrix X as follows:

$$\gamma = \frac{1}{N_{features} \cdot \sigma_X^2} \tag{4.3}$$

Question 4.3 [9 pts] In this part you will compare the models trained in previous sections based on the test set performance. Plot a boxplot with respect to the selected metric and discuss your results. Determine which models performed best and worst. Would you expect an increase in performance if SVM was trained directly on image pixels? Indicate the effect of γ and C on the decision boundary of SVM. How will the model perform on training and test datasets if is increased without bounds. Discuss your results.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.