The University of Texas at Austin

EE382V Scalable Machine Learning — Fall 2019

HOMEWORK ONE

Dimakis Due: Sunday, September 22, 3am.

Upload a pdf of your report with all the written comments that also includes screenshots from Kaggle private leader board. Upload also a separate zip file with all the code needed to replicate your results in the homework report. Please start early. During Sept 14-15 lectures we will experiment with improving the model performance in question 3, so go through the first steps.

Problem 1: Linear Algebra videos and warmup.

1. Watch: Vectors, what even are they? — Essence of linear algebra, Chapter 1 `https://www.youtube.com/watch?v=fNk_zzaMoSs`

2. Chapter 2 `https://www.youtube.com/watch?v=k7RM-ot2NWY`
   In this video, in Minute 2:51, two vectors are written by their coordinates. Compute the coordinates of the vector that is the sum of twice the first (left one) plus the second (right one).

   Are the vectors [1,1], [1,0], [0,1] linearly dependent?
   Write the third as a linear combination of the first two.

3. Chapter 3 `https://www.youtube.com/watch?v=kYB8IZa5AuE`
   At minute 6.36, a Matrix and a vector is given. Write down how this matrix transforms this vector. Based on what you learned, write a matrix that rotates the 2D space by 90 degrees clockwise.

4. Chapter 4 `https://www.youtube.com/watch?v=XkY2DOUCWMU`

   At 3.34 a composition matrix is shown. Apply this Composition linear transformation to the vector $[1, 2]^T$ and write the transformed vector.

5. The determinant — Essence of linear algebra, chapter 6 `https://www.youtube.com/watch?v=Ip3X9LOh2dk`
   In 9.40 there is a quiz question: Write your answer one sentence.

6. Dot products and duality — Essence of linear algebra, chapter 9 `https://www.youtube.com/watch?v=LyGKycYT2v0`
   Project the vector $[1, 2, 3]^T$ on the vector [1,1,1]. Write the projected vector.
   Project the vector $[1, 2, 3]^T$ on the span of the vectors $[1, 0, 0]^T$ and $[1, 0, 0]^T$. Write the projected vector.

Problem 2: Linear Algebra in Python. You can use all Python functions to solve this problem.

1. Consider the linear subspace $S = span\{v_1, v_2, v_3, v_4\}$ where $v1 = [1, 2, 3, 4], v2 = [0, 1, 0, 1], v3 = [1, 4, 3, 6], v4 = [2, 11, 6, 15]$. Create a vector inside S different from v_1, v_2, v_3, v_4. Create a vector not in S. How would you check if a new vector is in S?

2. Find the dimension of the subspace S.
3. Find an orthonormal basis for the subspace S.
4. Solve the optimization problem $min_{x \in S} ||x - z^*||_2$ where $z^* = [1, 0, 0, 0]$.
5. (Tricky) Is there a relation of this optimization problem with linear regression? Discuss.

Problem 3: Starting supervised learning in Kaggle.

1. Lets start with our first Kaggle submission in a playground regression competition. Make an account to Kaggle and find https://www.kaggle.com/c/house-prices-advanced-regression-techniques/
2. Follow the data preprocessing steps from https://www.kaggle.com/apapiu/house-prices-advanced-regression-techniques/regularized-linear-models.

   Then, split the data into a train and test set and fit a linear regression model (without any regularization) to make predictions. Report the performance (RMSE) of this model on the train set, the test set and on the Kaggle private leader board.
   (Hint: remember to exponentiate np.expm1(ypred) your predictions).
3. Fit a ridge regression (i.e. using $\ell - 2$ regularization) using $\alpha = 0.1$. Make a submission of this prediction. Again, report train error, test error and your score/position on Kaggle LB.
4. Train a ridge regression and a lasso regression model. Optimize the alphas using cross validation. What is the best score you can get from a single ridge regression model and from a single lasso model ? Report also the best hyperparameter α that you find.
5. Plot the l_0 norm (number of nonzeros) of the coefficients that lasso produces as you vary the regularization hyperparameter alpha.
6. Add the outputs of your models as features and train a ridge regression on all the features plus the model outputs (This is called Ensembling and Stacking). Be careful not to overfit. What score can you get?
7. Improve your perfomance by running any model or method you would like to try. Experiment with feature engineering and stacking many models. You are allowed to use any public tool in python. No nonpython tools allowed. Read the Kaggle forums and kernels to get ideas. Include in your report if you find something in the forums you like, or if you made a post or code, especially if other Kagglers used it afterwards.
8. Report the best RMSE you got and the position on the private Kaggle leader board, and how you got it.