PROBLEM SET 2: APPLIED MATHEMATICS 216

Due: Friday February 11 at 11:59pm

Goals for the week.

(1) Unsupervised learning:
   (a) Random Projections
   (b) Principal component analysis
   (c) Support Vector Machines

(2) Supervised problems: Classification with logistic regression and neural networks.

Problems.

(1) **Breast cancer classification**

In this problem, you will classify whether cells in the given breast cancer cell image are malignant or benign. There are 58 images in the dataset, together with their labels. The images are labeled 1 if they are malignant and 0 if they are benign.

First, take a look at the visualization provided in the notebook 'P1_breast_cancer' and get a feeling of the images we are working with. Can you identify any of the features in the dataset? Is there any noticeable difference between these two classes? (You do not need to provide answers to these two questions). To give you an idea behind the fancy models in the papers that Sherry discussed in class on Friday, here we will try to make a classification prediction ourselves using simple classification methods–Principal Component Analysis (PCA) and logistic regression.

(a) Fit a logistic regression model on the images and report the following error metrics: (i) Mean accuracy. (ii) Balanced accuracy. (iii) ROC curve with AUC on the test set data. You may want to play with different regularization[1]

(b) Do the above exercise again with downsampled images.

(c) ...and again with the top principle components. Change the number of principle components you use as input features and see how the performance changes.

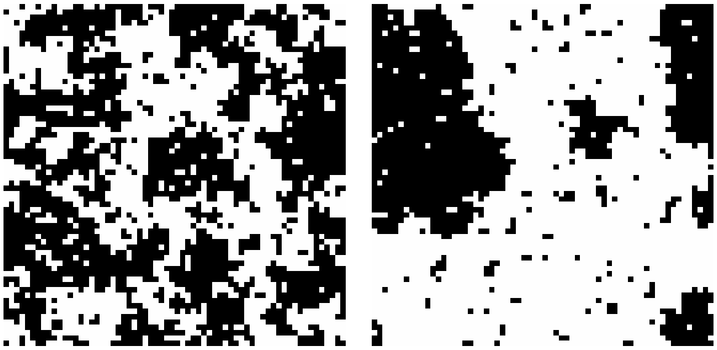
(d) Show scatter plots of top 2 principle components for both malignant and benign train set images.

(e) Discuss what you have learned! For example, does regularization or data processing steps affect classification result? Does the use of PCA as input features improve the classification? If so, then why?

[1] Note that the Google Machine Learning Crash Course materials assigned this week describes these accuracy metrics. Another excellent source is wikipedia, or the documentation in the sklearn package we use to compute these quantities.

(2) **The Ising Model**

Following our class and section discussions, let us consider simulations of the 2D Ising model. Here, we are providing two datasets (PS2-a-highT, PS2-a-lowT). The first has spin states at a temperature $T_1 > T_c$, the second at a temperature $T_2 < T_c$. The datasets each contain 500 examples of spins on a 64×64 grid. Note that T_1 and T_2 are closer to each other than the two temperatures within the section notebook. This can be seen within the images below with disordered ($T_1 > T_c$) being on the left, and ordered ($T_2 < T_c$) being on the right.



The transition from order to disorder is continuous and, thus, it is often difficult to tell states apart near the critical temperature. In class, we saw that Logistic Regression fails miserably on this system. Now you will try two other methods, and see how well they work.

Part (a): Try to classify the images using (1) an SVM and (2) a simple Neural Network.

(i) Import the images (PS2-a-highT, PS2-a-lowT). These images are 64×64.
(ii) Train your classifier (SVM & NN) on these images.
(iii) Play with the different options and hyper-parameters within your models to optimize your prediction.

As you will see, classifying images has become a relatively easy task with the tools we now have. What remains more difficult is using our tools to extract physics. Using our classifier from *Part (a)*, we are going to attempt to find the critical temperature (T_c) of our system.

Part (b): Estimate the critical temperature T_c using your SVM.

(i) Import the images (PS2-b-highT, PS2-b-lowT). These images are 32×32.
(ii) Train your classifier (SVM) on these images.
(iii) Generate images (of size 32×32) at different temperatures using the **Ising()** class provided in section. As always, feel free to modify any provided code. Using these images and your trained models, try and estimate the transition temperature T_c.

Now, we will try the even more difficult task of defining the temperature of the provided images (PS2-b-highT, PS2-b-lowT).

Part (c): Estimate the temperatures of the two provided datasets which you have trained on: 'PS2-b-highT' and 'PS2-b-lowT'.

You are free to try any method that you can imagine. There are many ways to attempt this problem. If you are unsure of what to do, you can always ask us for help! [Hint: An SVM will have more difficulty classifying images which are generated near the critical temperature. Try to take advantage of this fact.]

(3) **Convexity of Logistic Regression** We discussed in class that "a function is **convex** if the line segment between any two points on the graph of the function does not lie below the graph between these two points", or rather that the derivative is monotonically nondecreasing, or rather that $f''(t) > 0$. (Wikipedia). This week we have discussed three different optimization problems

(a) Regression through a L_2 loss (last week)
(b) Logistic Regression, through the cross entropy loss.
(c) Support Vector machines.

(a) Please comment on whether the losses for both linear regression and logistic regression are convex. You are free to use the internet, textbooks or whatever you find useful to help you with your argument.
(b) Why is convexity an important property for optimization problems?
(c) Give an example of a function whose loss is not convex.
(d) **Extra Credit:** Consider the cross entropy loss function for logistic regression, when evaluated on the training set for MNIST digits we discussed in lecture. Write a piece of code to plot this loss function as a function of one of the weights, with all of the other weights held fixed. Compare the shape of the function before and after training.
(e) **Extra Credit 2:** Repeat the above extra credit but for the cross entropy loss for the neural network, both before and after training. Comment on convexity.
(f) **Extre Credit 3:** Reconsider this entire problem for the loss function in support vector machines.

Submission Instructions.

(1) Submit 2 notebooks to Canvas:
LASTNAME_FIRSTNAME_P1
LASTNAME_FIRSTNAME_P2
(2) Submit your temperatures (T_1, T_2) to Kaggle