MGMTMSA 403: Optimization

Assignment 4: Logistic Regression and Gradient Descent

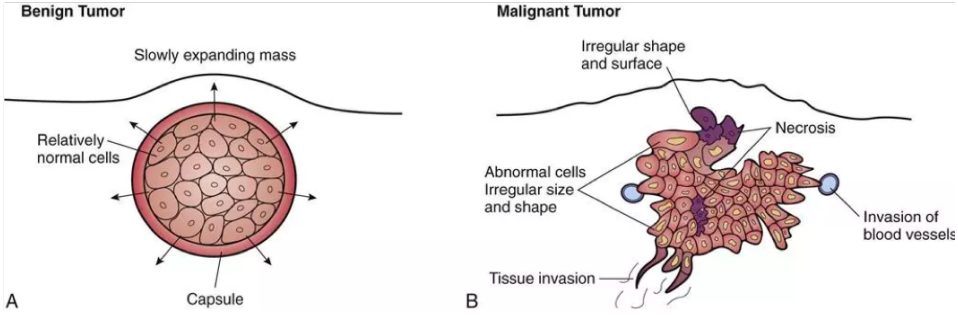
Due on BruinLearn by 11:59pm on February 14th.

Background

The file `LRTrain.csv` contains information from 300 images of malignant (i.e. cancerous) and benign (i.e., non-cancerous) breast tissue. The data set describes attributes of the cell nuclei in each image. Each row of the dataset corresponds to one image. For each image, ten different attributes related to the cell nuclei are recorded:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension (a measure of how "complex" the perimeter is)

Because each image contains multiple cell nuclei, three quantities are measured for each of the 10 attributes above: the mean, standard error, and worst case value. This results in a total of 30 features for each image. The goal of this assignment is to train a logistic regression classifier using gradient descent, which will then be used to predict whether or not each image was taken from cancerous tissue.



Questions

1. Train a logistic regression classifier using gradient descent on the training data set `LRTrain.csv`. The first 30 columns are feature data, and the 31st column is the label of the observation (1 = cancerous, 0 = not cancerous). As a starting point, try using a step size of $\gamma = 0.00001$ and a maximum of $T = 2000$ iterations. You can also use a termination criterion related to the norm of the gradient (`np.linalg.norm()` may be useful). Experiment with different step sizes and termination criteria to try to obtain a good model fit.

2. Once you have trained your logistic regression classifier, compute the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR) on the test dataset `LRTest.csv`. Re-calculate these performance metrics for each value of the threshold parameter t in $t \in \{0, 0.1, \ldots, 0.9, 1\}$. Report the performance of your classifier in the following table format:

t	TPR	FPR	TNR	FNR
0.0				
0.1				
0.2				
$\vdots$				
0.9				
1.0				

   The four performance metrics can be calculated as follows:

$$\text{TPR} = \frac{\text{\# of true positives}}{\text{\# of positives}}, \quad \text{FPR} = \frac{\text{\# of false positives}}{\text{\# of negatives}}$$

$$\text{TNR} = 1 - \text{FPR}, \quad \text{FNR} = 1 - \text{TPR}$$

   In the above, "# of positives" is the number of observations in which $y_i = 1$ in the test set, and "# of negatives" is the number of observations in which $y_i = 0$ in the test set. Similarly, "# of true positives" denotes the number of $y_i = 1$ observations in the test set that are correctly classified, and "# of false positives" is the number of $y_i = 0$ observations in the test set that are incorrectly classified.

HINT: You may find it useful to define three Python functions: `grad(w,x,y)` to compute the gradient at a given weight vector $\mathbf{w}$, `fval(w,x,y)` to compute the likelihood function at $\mathbf{w}$, and `gradnorm(w,x,y)` to evaluate the norm of the gradient at $\mathbf{w}$. See Lecture 5 slides for the mathematical expressions of these quantities.