UCSB Spring 2019

ECE 283: Homework 1

Topics: Classification using logistic regression
Assigned: Wednesday April 3
Due: Tuesday April 16 (electronic submission by 5 pm)
Reading: Posted notes on logistic regression; Bishop (Sections 4.3.2-4.3.4)

1) **Generating 2D synthetic data for binary classification**
Write Matlab or python code to generate samples from each class, as specified below. In order to visualize what the classes look like, do a scatterplot showing 200 samples each from class 0 and class 1.
Class 0: Gaussian with mean vector $\mathbf{m} = (0,0)^T$ and covariance matrix $\mathbf{C}$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 2$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
Class 1: Gaussian mixture with two components:
Component A: $\pi_A = \frac{1}{3}$, $\mathbf{m}_A = (-2,1)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs: $\lambda_1 = 2$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1/4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = -\frac{3\pi}{4}$.
Component B: $\pi_B = \frac{2}{3}$, $\mathbf{m}_A = (3,2)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs: $\lambda_1 = 3$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.

2) Assuming equal priors, implement the MAP decision rule, and classify the samples generated in part 1 using the rule. You will have to figure out how to display the decision boundary. Make sure you specify how you are computing the decision boundary in your report.

3) Estimate the conditional probability of incorrect classification for each class with the MAP decision rule using simulations. Choose the number of samples large enough to get good estimates: at least a factor of 10 larger than the inverse of the error probabilities you expect to get. *Save these samples* so you can do a direct comparison with logistic regression.

4) Generate N training data samples using your simulation model ($N/2$ from each class). You should play with the value of N to get “adequate” performance: start with $N = 200$, but then go down and up by factors of two. In order to make the comparison across different N easier, use a single set of training samples, and take the first $N/2$ data points from each class for each value of N that you consider. Using a Gaussian kernel $k(x,x') = \exp\left(-||x-x'||^2/2\ell^2\right)$, apply kernelized logistic regression with Newton's method to find a classifier. You will need to play with the hyperparameter ℓ. For the Newton iterations, note that you may either process the entire data set, or use the data sequentially in smaller batches (or even one data point at a time). (Do some digging on your own to find out what the best practices are.) Comment on what is the smallest N you can get away with, as well as the number of iterations you needed to run for convergence.

5) Plot the training data points and show the decision boundaries (again, you will have to figure out a good way to compute and display these, and should specify how you do it). Do you notice overfitting? (If so, you should do some ℓ_2 regularization).

6) Using the samples in part 3, compute empirical estimates for the conditional probability of incorrect classification for each class. Compare with the corresponding estimated probabilities for the MAP rule obtained in part 3.

7) Repeat parts 4 through 6 for (non-kernelized) logistic regression with explicit linear, quadratic and cubic features:

$$\phi(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3)$$

Do you notice overfitting? (If so, you should do some ℓ_2 regularization.) Comment on how the decision boundaries, convergence and misclassification probabilities compare to the kernelized solution.