# ECE 283: Homework 1

*Topics:* Generation of samples from mixture Gaussian distribution; Model-based classification; Data-driven classification using logistic regression
*Assigned:* Tuesday March 31
*Due:* Friday April 17 (electronic submission by 5 pm)
*Reading:* Handout 2 (model-based classification and Gaussian distributions) and Handout 3 (logistic regression), plus associated lecture videos as needed

**1) Generating 2D synthetic data for binary classification:** Write 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 $\mathbf{m} = (1, -1)^T$ and covariance matrix $\mathbf{C}$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
*Class 1:* Gaussian mixture with two components:
Component A: $\pi_A = \frac{2}{3}$, $\mathbf{m}_A = (-1, 0)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs: $\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1/2$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = -\frac{3\pi}{4}$.
Component B: $\pi_B = \frac{1}{3}$, $\mathbf{m}_B = (4, 1)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs: $\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.

**2) MAP decision rule:** Assuming equal priors, implement the MAP decision rule, and classify the samples generated in part 1 using the rule, showing the classification results in the 2D plot with the samples. 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) Inference performance of MAP rule:** 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** as test samples so you can do a direct comparison with logistic regression.

**4) Logistic regression with fixed feature vector:** 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. Train, using Newton's method, a classifier employing (non-kernelized) logistic regression with explicit linear, quadratic and cubic features:

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

(a) Plot the value of the cost function as a function of the number of iterations for different levels of $\ell_2$ regularization (i.e., for different values of the $\lambda$ parameter in the notes). Discuss the impact of $N$ and $\lambda$. Make sure you include $\lambda = 0$ (no regularization) among the cases you consider.

(b) Plot the training data points and show the decision boundaries for different values of $\lambda$. Do you notice overfitting for $\lambda = 0$? How do the results depend on $N$?

(c) 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. Again, comment on the impact of the number of training samples $N$ and the regularization parameter $\lambda$.

(d) Based on (a)-c), comment on what is the smallest number of training samples $N$ you can get away with, what is a good choice for the regularization parameter $\lambda$, and the number of Newton iterations needed.

**5) Kernelized logistic regression:** Now, train a classifier using kernelized logistic regression with $k(x, x') = \exp(-||x - x'||^2/2\ell^2)$. Repeat steps (a)-(d), but this time realize that you also need to play with the hyperparameter $\ell$.

**6) Comparison of non-kernelized and kernelized classifiers:** Comment on how the decision boundaries, convergence and misclassification probabilities compare for the best non-kernelized and kernelized classifiers you have found. Set $N$ to be the smallest number of training points that you have been able to get good inference results in both cases. (Note that the best choice of regularization parameter might be different in the two cases, and we have an additional hyperparameter $\ell$ in the kernelized setting).