# SD 675 Pattern Recognition

*Assignment 2*
(Labs are to be done individually. Do not write a formal report.)

## Purpose

This lab investigates orthonormal transformations and distance-based classification.

## Class Data

We will consider four cases. The first three are Gaussian, with the following given means and covariances:

1. $\mu_A = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mu_B = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
2. $\mu_A = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \Sigma_A = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix} \quad \mu_B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}$
3. $\mu_A = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \quad \mu_B = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 7 & -3 \\ -3 & 4 \end{bmatrix}$
4. See home page for Matlab file `assign2.mat`

In each case, each cluster has $N_A = N_B = 200$ data points. For MAP the clusters are equally likely.

## Generating Clusters

Use the Matlab function **randn** to assist in the generation of the 2D clusters for cases 1-3. The **randn** function will produce normally (ie, Gaussian) distributed data with mean 0 and variance 1.0. To create the correlated data as required, you will need to apply a transformation (think eigendecomposition) to the uncorrelated, equal-variance data (see chapter 3 of the SD372 notes).

## Distance Classifiers

We will be considering six classifiers:

1. Minimum Euclidean Distance (MED), with the sample mean as the prototype.
2. Minimum Generalized-Euclidean Distance (GED, also called MICD in the 372 notes), using *sample* means and covariances.
3. Nonparametric classifier NN using a Euclidean distance.
4. Nonparametric classifier 2-NN using a Euclidean distance.
5. Nonparametric classifier 4-NN using a Euclidean distance.
6. Although not distance-based, we will also show the MAP classifier as a reference, for the first three cases, using exact means and covariances.

For *each* of the four cases plot the class samples, the MED and GED classification boundaries, and for cases 1-3 the the unit standard deviation contours and MAP classification boundary, all superimposed on the same plot. (ie, four plots; one per case)

Also produce plots comparing the NN, 2-NN, and 4-NN classifiers for case 3. On each plot superimpose the optimal classification boundary (ie, MAP), computed from the exact means and covariances, and assuming the two classes to be equally likely. (ie, total of three plots)

Note that you should *not* try to find the boundaries analytically. Approach the problem numerically: grid the domain, classify each point, and then generate a contour plot (`help contour` in MATLAB).

Comment briefly on both sets of plots.