Lab1

MAP and Gaussian Datasets

Synthetic Gaussian datasets

- **Case 1:** $f(x \mid w_c) = N(\mu_c, \sigma^2 I)$

example with c=2 classes, d=3 features

in Lab0 code: Variables M_means and M_covar

- **Case 2:** $f(x \mid w_c) = N(\mu_c, C)$

example with c=4 classes, d=2 features

Apply MAP for cases 1 or 2:

- discriminant is a linear function
- decision boundaries are hyperplanes (lines for d=2, planes for d=3)
- decision boundary: $\boxed{w^t x + w_0 = 0}$

$$w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + w_0 = 0$$

Synthetic Gaussian datasets

- **Case 3:** $f(x \mid w_c) = N(\mu_c, C_c)$

in Lab0 code: Variables M_means and M_covar

example with c=4 classes, d=2 features

Apply MAP for case 3:

- discriminant is a quadratic function
- decision boundaries are hyper-quadratic (lines, spheres, ellipsoids, parabolodis, hyperboloids...)
- decision boundary:

$$w^t A x + w^t x + w_0 = 0$$

example d=2 $$x_1^2 A_{11} + x_2^2 A_{22} + x_1 A_{12} x_2 + x_2 A_{21} x_1 + w_1 x_1 + w_2 x_2 + w_0 = 0$$

Synthetic Gaussian datasets

Obtaining a quadratic classifier using Matlab:

- Classifier design using the training dataset: features in matrix X, labels in vector Label

```
quaclass = fitcdiscr(X,Labels,'discrimType','quadratic');
```

- Parameters of the boundary between class1 and class2:

```
w_0 = quaclass.Coeffs(class1,class2).Const;
w = quaclass.Coeffs(class1,class2).Linear;
A = quaclass.Coeffs(class1,class2).Quadratic;
```

- Predicted labels for data in matrix X

```
Labels_predict = predict(quaclass,X);
```

Parameters analized

- **Mahalanobis distance**: squared Mahalanobis distance between an observation x and a sample distribution with mean μ_2 and covariance matrix C_2:

$$D_M(x, x \in \omega_2) = (x - \mu_2)^T C_2^{-1} (x - \mu_2)$$

  squared Mahalanobis distance between vectors from two classes:

$$D_M(x \in \omega_1, x \in \omega_2) = \sum_{x \in \omega_1} (x - \mu_2)^T C_2^{-1} (x - \mu_2)$$

  where the mean and covariance matrix are estimated using the training dataset. Note that this is not a symmetric measure

- **Scatter plot**

- **Confusion matrix**

- **Receiver Operating Characteristic (ROC)**