

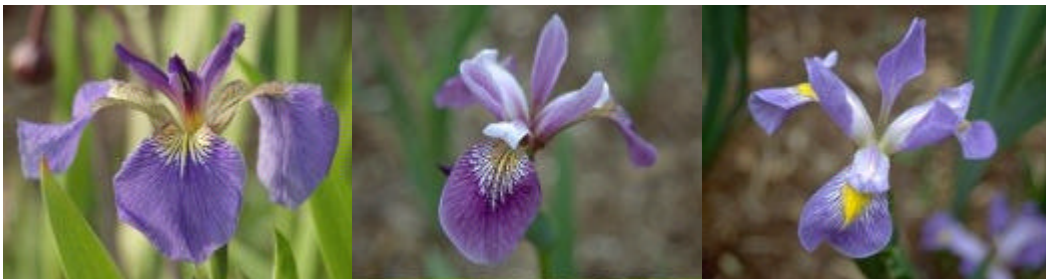
Práctica 1

- **Objetivo**

Estudiar de forma práctica el modelo bayesiano de clasificación y el aprendizaje basado en la estimación por máxima verosimilitud aplicada a un problema de clasificación.

- **Problema a Resolver**

Clasificar tres especies de flores (Iris setosa, versicolor y virginica) utilizando la longitud y anchura de los sépalos (Conjunto de las iris de Fisher).



Iris setosa, versicolor y virginica

- **Metodología**

Estimación por máxima verosimilitud en el caso gaussiano.

- **Contenido de la práctica.**

1. Estimar por máxima verosimilitud los parámetros de la distribución normal para cada clase
2. Calcular los coeficientes de la función discriminante en función de las muestras de cada clase y el error de clasificación.

- Inicializar la librería:
`sprint(INIT_SILENT);`

- Cargar los conjuntos de datos con:
`DATASET* dset = load_dataset(char *filename);`
con:
dset: conjunto de datos
filename: nombre del fichero

En `dset->NumInputs` está el número de características
En `dset->NumOutputs` está el número de clases

El fichero `fisher1.txt` tiene 102 datos para entrenamiento (34 por clase) y el `fishert.txt` 48 datos para testeo (16 por clase) y se utiliza para calcular el error.

- Calcular las clases presentes en el conjunto de datos con:

```
int* labels = GetLabels(DATASET* dset);
con:
dset: conjunto de datos
labels: vector de enteros. Su reserva de memoria se hace con la función ivector
```

- Calcular los estimadores de máxima verosimilitud (media y matriz de covarianzas) para el conjunto de datos e imprimirlos con:

```
int ret = mean_var_dataset(DATASET* dset, int class, double* mean, double** cov)
con:
dset: conjunto de datos
class: clase para la que calcular el estimador
mean: vector de medias. Su reserva de memoria se hace con la función vector
cov: matriz de covarianzas. Su reserva de memoria se hace con la función matrix
```

Para imprimir el vector de medias y la matriz de covarianzas se sugiere utilizar las funciones `fprintf_vector` y `fprintf_matrix`

- Calcular las funciones discriminantes y el error de clasificación con:

```
double calc_multiclass_error(DATASET* lset, DATASET* tset, int trace, int function);
```

```
lset: conjunto de entrenamiento
tset: conjunto de testeo.
trace=1: imprime las funciones discriminantes
function=1: usa el conjunto de aprendizaje para el cálculo del error. En otro caso usa el conjunto de testeo.
```

devuelve: una estimación de la probabilidad de error (frecuencia relativa de muestras mal clasificadas del conjunto de testeo) o un número negativo en caso de error.

- Liberar los datos y cerrar la librería:

```
Los vectores y matrices se liberan con free_ivector, free_vector y free_matrix
delete_dataset(dset);
spxexit(EXIT_SILENT);
```

Nota: La función `calc_multiclass_error` necesita las rutinas `matinvd`, `lubksb` y `ludcmp` de las "Recetas numéricas en C" y se proporcionan en el fichero de la práctica

Todo el material necesario se puede descargar de la página web con la práctica.