PMR Assignment 2018

Data file details

This archive contains 4 data files

- `diag_covar.npz` : Compressed Numpy file containing variables related to the diagonal covariance matrix Gaussian approximation to the posterior density across player skills.
- `full_covar.npz` : Compressed Numpy file containing variables related to the full covariance matrix Gaussian approximation to the posterior density across player skills.
- `diag_covar.mat` : MATLAB workspace file containing variables related to the diagonal covariance matrix Gaussian approximation to the posterior density across player skills.
- `full_covar.mat` : MATLAB workspace file containing variables related to the full covariance matrix Gaussian approximation to the posterior density across player skills.

The variables defined in each file are detailed in the assignment handout.

Language for assignment

You are free to use whichever programming language you are most comfortable with to complete the coding sections of this assignment. The main requirements are that is able to handle 1D and 2D array data and can produce good quality plot outputs. Two suggested options are

- **Python + Numpy**
  Python is a free, open source general purpose interpreted programming language. It is included by default with most Linux distributions and can be easily and freely installed on most operating systems. On Informatics DICE systems it can be loaded by running `python` from a terminal. Alternatively running `ipython` will load the enhanced interactive IPython interpreter and `ipython notebook` the web-based notebook interface. Numpy is a scientific computing package in Python, defining a flexible N-dimensional array object and various mathematical and linear algebra routines. Scipy, a closely related library, offers further useful functionality particularly in the `scipy.stats` module. Both `numpy` and `scipy` are available in the DICE Python installation.
- **MATLAB**
  MATLAB is a numerical computing environment and programming language particularly aimed at performing linear algebra like operations with large arrays of numerical data. It is a commercial product and is not freely available to download, however it is available on Informatics DICE systems

- either run `matlab -desktop` from a terminal or select MATLAB from the applications menu.

The data has been provided in formats which are easy to load in Python and MATLAB, however other languages may also support loading these formats. For example the statistics oriented language R has a package `R.matlab` which can be used to load `.mat` files.

Loading the data

Python + Numpy

The `.npz` files can be loaded using the `numpy.load` function. This returns a dictionary-like object with the variable names acting as the keys. So for example starting a `python` interpreter in the same directory as the data files and running

```
import numpy as np
diag_covar = np.load('diag_covar.npz')
```

will mean `diag_covar` is now a dictionary-like object which can be used to access the approximation density variables. For instance

```
print(diag_covar['approx_mean'])
```

will print the approximation density mean vector to the interpreter output.

If you wish to load all of the variables in the loaded dictionary-like object in to the global namespace you can use

```
globals().update(diag_covar)
```

At which point you will be able to then directly use variable names e.g.

```
print(approx_mean)
```

A more complete example of loading the data and performing some operations with it is given below.

```
import numpy as np
import scipy.stats as stats

# create pseudo-random number generator object
seed = 1234
prng = np.random.RandomState(seed)

# load data file (np.load returns dict like object)
diag_covar = np.load('diag_covar.npz')

# extract some of variables from loaded dict
approx_mean = diag_covar['approx_mean']
```

```
approx_covar = diag_covar['approx_covar']  # vector for diag_covar
n_players = diag_covar['n_players']

# draw a random skill sample from Gaussian posterior approximation
skill_sample = (approx_mean +
                prng.normal(size=n_players) * approx_covar**0.5)

# calculate log probability density at sample point
log_pdf_sample = stats.multivariate_normal.logpdf(
    skill_sample, approx_mean, approx_covar)
```

MATLAB

The `.mat` files can be loaded using the inbuilt `load` function. This populates the current workspace with the variables stored in the data file. In a MATLAB command window with the current directory set to the directory the extracted data files are in, running

```
load('diag_covar.mat')
```

will load all of the variables in the data file into the workspace. Alternatively `.mat` data files can be loaded using the MATLAB *Open File* dialog.

The following will perform a roughly equivalent set of operations to the Python example above.

```
% load data file objects in to workspace
load('diag_covar.mat')

% draw a random skill sample from Gaussian posterior
skill_sample = approx_mean + randn(1, n_players) .* approx_covar.^0.5;

% calculate log probability density at sample point
pdf_sample = mvnpdf(skill_sample, approx_mean, approx_covar);
log_pdf_sample = log(pdf_sample);
```