SamplingMethod

April 10, 2022

[1]:
```
import jax.numpy as jnp
import jax.random as random
from scipy.spatial.distance import cdist                                    # cdist is␣
 ↪used for generating random covariance matrix
```

[2]:
```
from sklearn.preprocessing import PowerTransformer                          # For␣
 ↪Converting Uniform Distribution to Normal Distribution
```

[3]:
```
# Setting key for randomization in JAX
key = random.PRNGKey(3)
```

1 Sampling Function For MVN

[4]:
```
def mvn_samples(mu, cov, n_samples=10):
    """
    Arguments :
        mu    -- mean of shape (dimension)
        cov   -- covariance matrix of shaoe( dimension,dimension)
        n_sample -- default to 10
    output :
        samples = mvn_sample of shape (dimensions,n_samples)

    """
    pt = PowerTransformer()
    key = random.PRNGKey(3)
    d = cov.shape[0]  # number of dimensions

    U = random.uniform(key,[d,n_samples])

    U = pt.fit_transform(U.T)


    L = jnp.linalg.cholesky(cov)
    samples = L.dot(U.T) + mu

    return samples
```

2 Evaluating Sampling Function

[5]:
```
dimension = 2         # Change it if you want
n_samples = 10        # Change it if you want

# Mean Matrix
mu = random.uniform(key,(1,dimension))

# Covariance matrix
var_matrix = random.uniform(key,[dimension,1])
cov = jnp.exp(-cdist(var_matrix , var_matrix, "euclidean")) + 1e-7*jnp.
 ↪eye(dimension)
```

[6]:
```
# Generating Samples of MVN

X = mvn_samples(mu.T, cov, n_samples = n_samples)
```

[7]:
```
X.shape
```

[7]:
```
(2, 10)
```

[8]:
```
print(X)
```

```
[[-0.0850457   0.9052321   1.8278908  -1.013562    1.5303793  -0.19038439
  -0.17757845 -0.91956985  1.1277742   1.5269079 ]
 [ 0.27497375  0.6859927   2.0552926  -0.7787179   1.1575104   0.05094829
   0.30503133 -0.30214292  0.65278494  1.0617568 ]]
```

[8]:
```

```

3 Checking Means of MVN and Generated Samples

[9]:
```
print("Random Uniform Mean     : ", mu)
print("Samples Generated Mean  : ", mu)
```

```
Random Uniform Mean     :  [[0.4532044 0.516343 ]]
Samples Generated Mean  :  [[0.4532044 0.516343 ]]
```

Minor difference is Neglizible

[ ]:
```

```