project 15: more on Markov chains

task 15.1: products of column stochastic matrices and vectors

Recall that a column stochastic matrix is a square matrix $\boldsymbol{P} \in \mathbb{R}^{m \times m}$ where

$$P_{ji} \geq 0 \qquad \text{and} \qquad \sum_{j=1}^{m} P_{ji} = 1$$

and that an m-dimensional stochastic vector is a vector $\boldsymbol{u} \in \mathbb{R}^{m}$ where

$$u_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{m} u_i = 1$$

Now, given $\boldsymbol{P}$ and $\boldsymbol{u}$ as above, prove that the vector

$$\boldsymbol{v} = \boldsymbol{P}\,\boldsymbol{u}$$

is a stochastic vector, too.

task 15.2: products of column stochastic matrices

Prove that, if $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{m \times m}$ are columns stochastic matrices, then

$$\boldsymbol{R} = \boldsymbol{P}\,\boldsymbol{Q}$$

is a column stochastic matrix, too.

please turn page . . .

task 15.3: estimating Markov chains

Load the 3D data points in the file `q3dm1-path1.csv` into a data matrix $\boldsymbol{X}$. Using python's `numpy` module, this can be accomplished like so

```
import numpy as np
matX = np.loadtxt('q3dm1-path1.csv', delimiter=',')
matX = matX.T
```

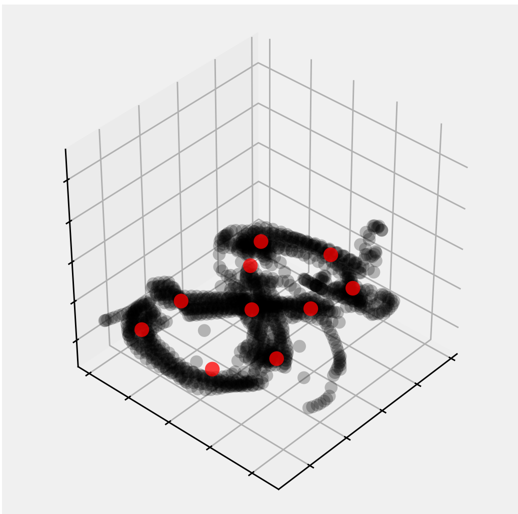
Note: if you know what you are doing (in the next couple of computational steps), then transposition of the array `matX` is not really necessary. However, as your instructor likes his matrices to be column matrices, all the following code examples assume `matX` has been transposed.

Now, cluster the 1288 columns of $\boldsymbol{X}$ into $k = 10$ clusters. If you do this using

```
import scipy.cluster.vq as vq
matM, inds = vq.kmeans2(matX.T, k=10, iter=100, minit='++')
matM = matM.T
```

you will obtain a matrix $\boldsymbol{M} \in \mathbb{R}^{3\times 10}$ whose columns $\boldsymbol{m}_i$ contain the cluster prototypes.

Plot the data in $\boldsymbol{X}$ and the prototypes in $\boldsymbol{M}$. Your result should look something like this



If it does not, then maybe you need to run `kmeans2` several times until your result is satisfactory.

Now, recall that the 3D data points in `q3dm1-path1.csv` resulted from recording movements of a human player on the Quake III map q3dm1.
In other words, we can understand the columns $\boldsymbol{x}_t$ of matrix $\boldsymbol{X}$ as the elements $\boldsymbol{x}[t]$ of a sequence of locations on that map.
Here is a crazy idea: We can also understand the columns $\boldsymbol{m}_i$ of matrix $\boldsymbol{M}$ as 3D representations of the states of a discrete state space

$$S = \{s_1, s_2, \ldots, s_k\}$$

such that

$$\begin{aligned} s_1 &\Leftrightarrow \boldsymbol{m}_1 \\ s_2 &\Leftrightarrow \boldsymbol{m}_2 \\ &\vdots \\ s_k &\Leftrightarrow \boldsymbol{m}_k \end{aligned}$$

If so, the we can turn the sequence

$$\boldsymbol{X} = \boldsymbol{x}[1] : \boldsymbol{x}[2] : \boldsymbol{x}[3] : \cdots \quad \in \mathbb{R}^{3^*}$$

into a sequence

$$\boldsymbol{s} = s[1] : s[2] : s[3] : \cdots \quad \in S^*$$

where

$$s[t] = s_b$$

and the index of *best matching unit* is

$$b = \underset{i}{\operatorname{argmin}} \left\| \boldsymbol{x}[t] - \boldsymbol{m}_i \right\|^2$$

Implement this idea! That is, write code that takes the columns $\boldsymbol{x}[t]$ of $\boldsymbol{X}$ and produces a sequence $\boldsymbol{s}$ of states $s[t] \in S$.

Next, given $\boldsymbol{s}$, estimate a Markov chain λ that most likely has produced this sequence. That basically is, estimate the corresponding matrix $\boldsymbol{P}_\lambda \in \mathbb{R}^{k \times k}$

of state transition probabilities. Recall that maximum likelihood estimates of the entries of $\boldsymbol{P}_\lambda$ are given by

$$\left(\boldsymbol{P}_\lambda\right)_{ji} = \frac{n_{ji}}{\sum_{j=1}^{k} n_{ji}}$$

where n_{ji} counts the number of transitions $s_i \rightarrow s_j$ in sequence $\boldsymbol{s}$.

If you implement this procedure using `numpy`, you will obtain an array, say, `matP`. Print this array using

```
print (np.round(matP, 2))
```

Discuss what do you observe? Your result should be a matrix of a (more or less) special structure ...

Finally, repeat all of this for different choices of k. Also, it might be a good idea to warm up to the data in file `q3dm1-path2.csv`. So, do all of the above once again but for the data in `q3dm1-path2.csv`.