Homework 2

1. (a) for volume of high dimensional ball can be written as

$$\mathrm{V} = C_d r^d$$

Where C_d is coefficient that corresponded to dimension d and r is the radius.

Since the data points are uniformly distributed, the probability can be written as

$$\Pr\left(\left|\left|\mathrm{X}\right|\right|_2 \leq 1 - \varepsilon\right) = \frac{C_d(1 - \varepsilon)^d}{C_d \cdot 1^d} = (1 - \varepsilon)^d$$

Therefore, according to above formula, when d is a large number $(1 - \varepsilon)^d$ can be very small. Let $\mathrm{d} = \frac{\mathrm{k}}{\varepsilon}$,

$$\Pr\left(\left|\left|\mathrm{X}\right|\right|_2 \leq 1 - \varepsilon\right) = \lim_{\varepsilon \to 0}(1 - \varepsilon)^{\frac{\mathrm{k}}{\varepsilon}} = e^{-k}$$

So, when d is super large (meanwhile ε is super small), the probability is very small (when k is 10, it's around 0.000045).

$$\Pr\left(\left|\left|\mathrm{X}\right|\right|_2 > 1 - \varepsilon\right) = 1 - (1 - \varepsilon)^d$$

Then the probability that radius is more than $1 - \varepsilon$ is around 1.

In other words, most data points are close to surface.

(b) based on (a), for each point that have length over $1 - \mathrm{clnd}/\mathrm{d}$ is

$$\Pr\left(\left|\left|\mathrm{X}\right|\right|_2 > 1 - \frac{\mathrm{clnd}}{\mathrm{d}}\right) = 1 - (1 - \frac{\mathrm{clnd}}{\mathrm{d}})^d$$

denoted as P.

Therefore for d^{100} points that all satisfy such condition is

$$\Pr(\text{all data points satisfy}) = P^{d^{100}} \geq 0.99$$

Therefore, we have

$$(1 - \left(1 - \frac{\text{clnd}}{\text{d}}\right)^{d})^{d^{100}} \geq 0.99$$

$$\Rightarrow d^{100} \ln\left(\left(1 - \left(1 - \frac{\text{clnd}}{\text{d}}\right)^{d}\right)\right) \geq ln0.99$$

$$\Rightarrow 1 - \left(1 - \frac{\text{clnd}}{\text{d}}\right)^{d} \geq e^{\frac{ln0.99}{d^{100}}}$$

$$\Rightarrow 1 - \frac{\text{clnd}}{\text{d}} \leq e^{\frac{\ln(1-e^{\frac{ln0.99}{d^{100}}})}{d}}$$

$$\Rightarrow \text{c} \geq \frac{\text{d}}{\text{lnd}}(1 - e^{\frac{\ln\left(1-e^{\frac{ln0.99}{d^{100}}}\right)}{d}})$$

Also we can notice that $0 \leq 1 - \frac{\text{clnd}}{\text{d}} \leq 1 \Rightarrow 0 \leq c \leq \frac{d}{lnd}$

Therefore, c can be any number such that

$$\frac{\text{d}}{\text{lnd}}(1 - e^{\frac{\ln\left(1-e^{\frac{ln0.99}{d^{100}}}\right)}{d}}) \leq c \leq \frac{d}{lnd}$$

2. (a)

$$\text{Cov}(\text{X}, \text{Y}) = \text{Cov}(\text{X}, \text{aX} + \text{b})$$

$$= \text{Cov}(\text{X}, \text{aX}) + \text{Cov}(\text{X}, \text{b})$$

$$= aCov(X, X) + \text{Cov}(\text{X}, \text{b}) = aVar(X) + \text{Cov}(\text{X}, \text{b})$$

And we have

$$\text{Cov}(\text{X}, \text{b}) = \text{E}(\text{bX}) - \text{E}(\text{X})\text{E}(\text{b}) = \text{bE}(\text{X}) - \text{bE}(\text{X}) = 0$$

Hence, $\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = aVar(X)$

(b) for covariance of Y, we can obtain as follows

$$\mathrm{Var}(\mathrm{Y}) = \mathrm{Cov}(\mathrm{aX}+\mathrm{b}) = a^2Var(X)$$

Therefore, the correlation of X and Y should be

$$\mathrm{r} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} = \frac{aVar(X)}{\sqrt{Var(X)\cdot a^2Var(X)}} = 1$$

3. For this question, let $Y = X^2$, then we have

$$\Pr(\mathrm{Y}=0) = \frac{1}{3}, \Pr(Y=1) = \frac{2}{3}$$

Then we can calculate covariance of X and Y, that is

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = \mathrm{E}(\mathrm{XY}) - \mathrm{E}(\mathrm{X})\mathrm{E}(\mathrm{Y})$$

and

$$\begin{aligned}\mathrm{E}(\mathrm{XY}) = 1\cdot\Pr(\mathrm{X}=1,\mathrm{Y}=1) + 0\cdot(\Pr(\mathrm{X}=0,\mathrm{Y}=1)\\ + \Pr(\mathrm{X}=-1,\mathrm{Y}=0) + \Pr(\mathrm{X}=0,\mathrm{Y}=0)\\ + \Pr(\mathrm{X}=1,\mathrm{Y}=0)) + (-1)\cdot\Pr(\mathrm{X}=-1,\mathrm{Y}=1)\end{aligned}$$

Since $Y = X^2$, $(\mathrm{X}=0,\mathrm{Y}=1)$, $(\mathrm{X}=-1,\mathrm{Y}=0)$, $(\mathrm{X}=1,\mathrm{Y}=0)$ cannot happen, thus their probability is 0. And

$$\Pr(\mathrm{X}=1,\mathrm{Y}=1) = \Pr(X=1) = \frac{1}{3}$$

$$\Pr(\mathrm{X}=0,\mathrm{Y}=0) = \Pr(X=0) = \frac{1}{3}$$

$$\Pr(\mathrm{X}=-1,\mathrm{Y}=1) = \Pr(X=-1) = \frac{1}{3}$$

Therefore, $\mathrm{E}(\mathrm{XY}) = 0$

Similarly, we can have E(X)=0, E(Y)= $\frac{2}{3}$ by

$$\mathrm{E(X)} = 1 \cdot \Pr(\mathrm{X} = 1) + 0 \cdot \Pr(\mathrm{X} = 0) + (-1) \cdot \Pr(\mathrm{X} = -1) = 0$$

$$\mathrm{E(Y)} = 1 \cdot \Pr(\mathrm{Y} = 1) + 0 \cdot \Pr(Y = 0) = \frac{2}{3}$$

Thus, we obtain that

$$\mathrm{Cov(X, Y)} = \mathrm{E(XY)} - \mathrm{E(X)E(Y)} = 0$$

$$\Longrightarrow \mathrm{r} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} = 0$$

So Y and X is unrelated.

And for the first question, we can conclude that deterministic relationships don't imply correlation.

4. From the lecture, we have the way to update mean of the sequence, which can be written as

$$\overline{x_n} = \overline{x_{n-1}} + \frac{x_n - \overline{x_{n-1}}}{n} \quad (1)$$

where $\overline{x_n}$ is mean of first n elements, x_n is the nth element.

Let ${s_n}^2 = Var(x_1, x_2, \dots, x_n)$, then we have

$${s_n}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x_n})^2 \quad (2)$$

Thus, we can let ${d_n}^2 = (n-1){s_n}^2 = \sum_{i=1}^{n}(x_i - \overline{x_n})^2$, we divide the left side, that is

$${d_n}^2 = \sum_{i=1}^{n}(x_i - \overline{x_n})^2 = \sum_{i=1}^{n}{x_i}^2 + \sum_{i=1}^{n}\overline{x_n}^2 - 2\overline{x_n}\sum_{i=1}^{n}x_i$$

$$= \sum_{i=1}^{n}{x_i}^2 + n\overline{x_n}^2 - 2\mathrm{n} \cdot \overline{x_n}^2 = \sum_{i=1}^{n}{x_i}^2 - \mathrm{n} \cdot \overline{x_n}^2 \quad (3)$$

Similarly, we can have

$$d_{n-1}{}^2 = \sum_{i=1}^{n-1} x_i{}^2 - (\mathrm{n}-1)\cdot \overline{x_{n-1}}^2 \qquad (4)$$

Let (3)-(4),

$$d_n{}^2 - d_{n-1}{}^2 = \sum_{i=1}^{n} x_i{}^2 - \mathrm{n}\cdot \overline{x_n}^2 - (\sum_{i=1}^{n-1} x_i{}^2 - (\mathrm{n}-1)\cdot \overline{x_{n-1}}^2)$$

With (1),(2),(3), we can eventually have

$$d_n{}^2 = d_{n-1}{}^2 + (x_n - \overline{x_{n-1}})(x_n - \overline{x_n})$$

Thus for variance, we can use $d_n{}^2 = (n-1)s_n{}^2$ to get,

$$s_n{}^2 = \frac{n-2}{n-1} s_{n-1}{}^2 + \frac{1}{n-1}(x_n - \overline{x_{n-1}})(x_n - \overline{x_n})$$

Hence let $v_t = s_t{}^2$

$$v_t = \frac{t-2}{t-1} v_{t-1} + \frac{1}{t-1}(x_t - \overline{x_{t-1}})(x_t - \overline{x_t})$$

In a nutshell, my online algorithm for computing variance is

(1) $v_1 = 0,\ \overline{x_1} = x_1$

(2) Update mean at time t first with $\overline{x_t} = \overline{x_{t-1}} + \frac{x_t - \overline{x_{t-1}}}{t}$

(3) Update variance at time t with

$$v_t = \frac{t-2}{t-1} v_{t-1} + \frac{1}{t-1}(x_t - \overline{x_{t-1}})(x_t - \overline{x_t})$$

5. Similar to what we've taught, the algorithm is

   (1) $z_1 = x_1$, denominator $\mathrm{M} = f(x_1)$

   (2) For t=2,3,…

      a. Get x_t

      b. Update z_t with $\Pr(z_t = x_t) = \frac{f(x_t)}{f(x_1)+f(x_2)+\cdots+f(x_t)}$

c. Update M with $\mathrm{M} = \mathrm{M} + f(x_t)$

(this is to maintain denominator at all times)

Here to show why this is correct:

Pick any time t and any x_j such that $\mathrm{j} \leq \mathrm{t}$

and

$$\Pr(z_t = x_j)$$

$$= \Pr(\text{picked } x_j \textit{ at time } j) \cdot \Pr(\text{picked } \textit{nothing after time } j)$$

$$= \frac{f(x_j)}{f(x_1) + f(x_2) + \cdots + f(x_j)} \cdot \frac{f(x_1) + f(x_2) + \cdots + f(x_j)}{f(x_1) + f(x_2) + \cdots + f(x_{j+1})}$$

$$\cdot \frac{f(x_1) + f(x_2) + \cdots + f(x_{j+1})}{f(x_1) + f(x_2) + \cdots + f(x_{j+2})} \cdot \ldots$$

$$\cdot \frac{f(x_1) + f(x_2) + \cdots + f(x_{t-1})}{f(x_1) + f(x_2) + \cdots + f(x_t)}$$

$$= \frac{f(x_j)}{f(x_1) + f(x_2) + \cdots + f(x_t)}$$

Thus we have

$$\Pr(z_t = x_j) = \frac{f(x_j)}{f(x_1) + f(x_2) + \cdots + f(x_t)} = \frac{f(x_j)}{M}$$

Hence, we conclude that $\Pr(z_t = x_j) \propto f(x_j)$

6. (1)

The optimal solution with k=3 is that the centers are

$$\mu_1 = -9, \mu_2 = 0, \mu_3 = 9$$

The cost should be

$$\text{cost}(\mu_1, \mu_2, \mu_3) = \sum_{i=1}^{5} \min \left|\left|x_i - \mu_j\right|\right|^2$$

$$= 4$$

(2) suppose initial centers are

$$\mu_1 = -10, \mu_2 = -8, \mu_3 = 0$$

Then the points closest to μ_1 is -10, closest to μ_2 is -8 and closest to μ_3 is 0, 8, 10.

So after the first step,

$$\mu_1 = -10, \mu_2 = -8, \mu_3 = 6$$

And the points closest to μ_1 is -10, closest to μ_2 is -8 and closest to μ_3 is 0, 8, 10. Then we find it convergence.

So the final set of centers are

$$\mu_1 = -10, \mu_2 = -8, \mu_3 = 6$$

And the cost is 56

7. (1)

With the definition of k-centers algorithm, the optimal solution should be that with centers

$$\mu_1 = (1.5,1.5),\ \mu_2 = (1,8),\ \mu_3 = (5.5,1)$$

With cost

$$\text{cost}(\mu_1, \mu_2, \mu_3) = \sum_{j=1}^{3} \max_i \min_{1 \le j \le 3} d(x_i, \mu_j) = \frac{\sqrt{2}+1}{2}$$

(2)

Suppose we run farthest-first traversal on this data set, and select

$$\mu_1 = (1,1)$$

And we know dataset are

$$S = \{(1,1), (2,1), (5,1), (6,1), (1,2), (2,2), (1,8)\}$$

The farthest from μ_1 is (1,8) then select

$$\mu_2 = (1,8)$$

Again, the point farthest from μ_1 and μ_2 is (6,1), then we have

$$\mu_3 = (6,1)$$

Similarly, we can compute the cost

$$\text{cost}(\mu_1, \mu_2, \mu_3) = \sum_{j=1}^{3} \max_i \min_{1 \le j \le 3} d(x_i, \mu_j) = \sqrt{2} + 1$$

(3)

For a cluster, suppose the distance of optimal center and the farthest point is r, then the distance of two points in this cluster should be less than 2r, otherwise, it contradicts to the assume that the distance of optimal center and the farthest point is r. That's why the ratio can be guaranteed to less than 2.

And for the next question that there is no better algorithm if the centers are datapoints. Suppose there is, let say the ratio is α such that $\alpha < 2$. And if a point that is 2r from center should not be included in this cluster. In other words, this point should be included in other class. However,

in original clustering, this point is included in this cluster because it's closest to this specific center, if included in another cluster, the overall cost should increase. Then the overall ratio will increase, and eventually contradicts to our assumption.

Thus, our conclusion is proved. From the above, since we are selecting the datapoints as centers, it cannot avoid local optima.

CSE291 HW2

Coding

(after doing all the things, I restarted the kernel and ran all)

8

(a)

import all the packages that might be used

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
import scipy
from pylab import rcParams
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.cluster import KMeans
```

using numpy to upload the data we need

In [2]:
```
predicates = np.loadtxt('predicate-matrix-continuous.txt')
animals = np.loadtxt('classes.txt', dtype = 'str')[:, 1]
```

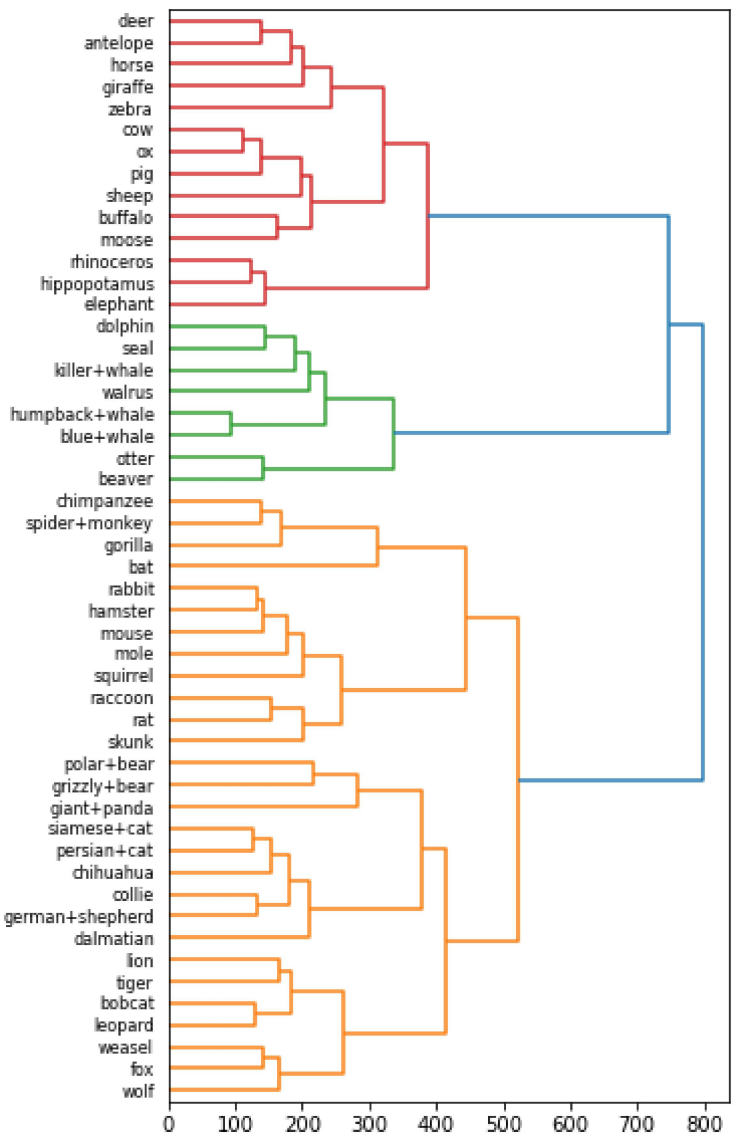
see what is going on

In [3]:
```
predicates
```

Out[3]:
```
array([[-1.  , -1.  , -1.  , ...,  2.35,  9.7 ,  8.38],
       [39.25,  1.39,  0.  , ..., 58.64, 20.14, 11.39],
       [83.4 , 64.79,  0.  , ..., 15.77, 13.41, 15.42],
       ...,
       [63.57, 43.1 ,  0.  , ..., 35.95, 28.26,  5.  ],
       [55.31, 55.46,  0.  , ...,  5.04, 18.89, 72.99],
       [10.22, 21.53, 27.73, ...,  3.96, 14.05, 37.98]])
```

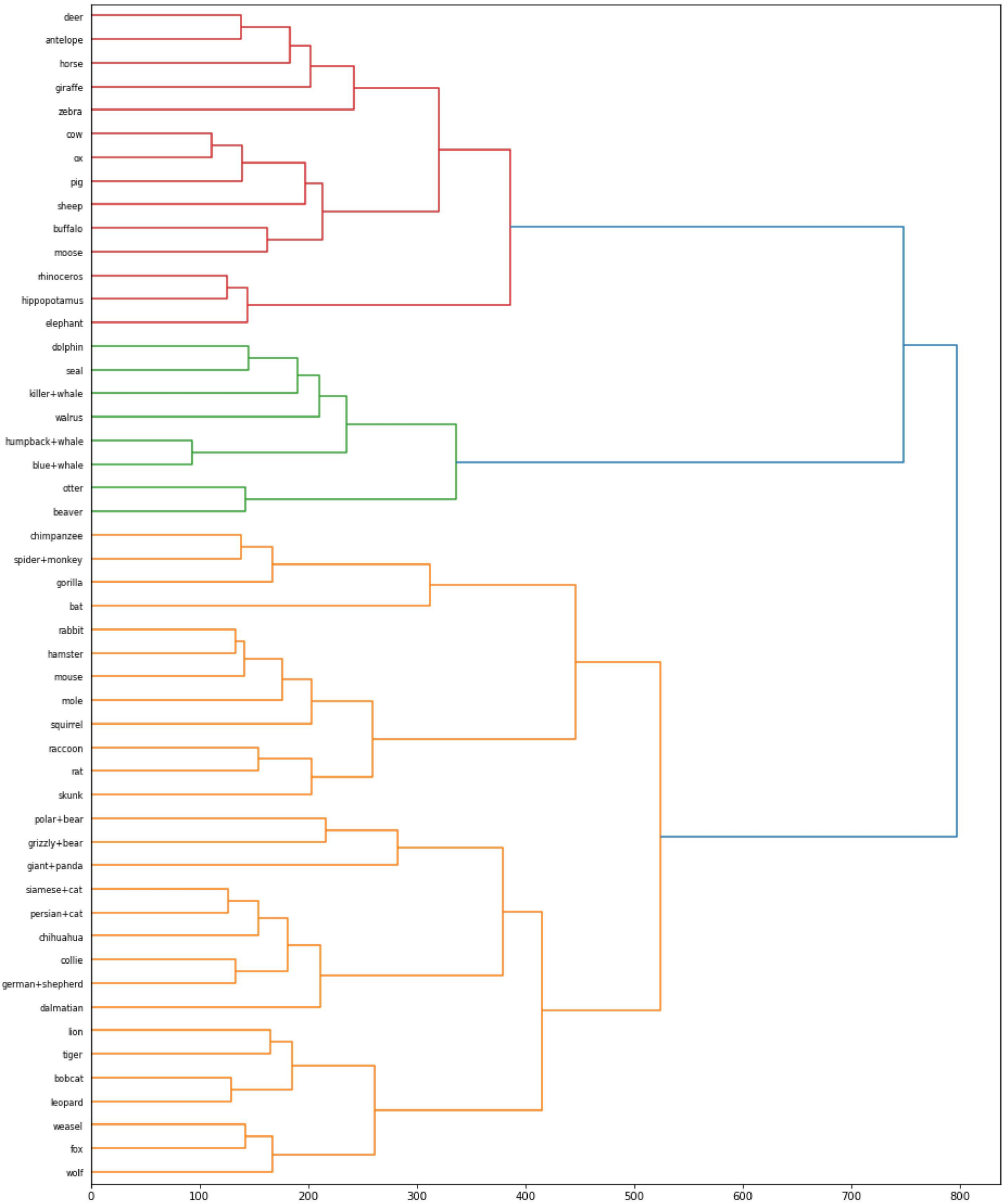
using linkage algorithm as instructed to cluster and display a dendrogram

In [4]:
```
lk = scipy.cluster.hierarchy.linkage(predicates, 'ward')
rcParams['figure.figsize'] = 5, 10
d_gram = dendrogram(lk, orientation = 'right', labels = animals)
```



for better show, reset the size to a better value and display again

In [5]:
```
rcParams['figure.figsize'] = 15,20
d_gram = dendrogram(lk, orientation = 'right', labels = animals)
```



(b)

now, we will show what specifically in each 10 cluster

In [6]:
```
k = 10
clusters = scipy.cluster.hierarchy.fcluster(lk, k, criterion = 'maxclust')

for i in np.unique(clusters):
    indices = np.where(clusters == i)[0]
    print('Cluster {}: '.format(i))
    print(animals[indices])
```

```
Cluster 1:
['tiger' 'leopard' 'fox' 'wolf' 'weasel' 'bobcat' 'lion']
Cluster 2:
['dalmatian' 'persian+cat' 'german+shepherd' 'siamese+cat' 'chihuahua'
 'collie']
Cluster 3:
['grizzly+bear' 'giant+panda' 'polar+bear']
Cluster 4:
['skunk' 'mole' 'hamster' 'squirrel' 'rabbit' 'rat' 'mouse' 'raccoon']
Cluster 5:
['spider+monkey' 'gorilla' 'chimpanzee' 'bat']
Cluster 6:
['beaver' 'otter']
Cluster 7:
['killer+whale' 'blue+whale' 'humpback+whale' 'seal' 'walrus' 'dolphin']
Cluster 8:
['hippopotamus' 'elephant' 'rhinoceros']
Cluster 9:
['moose' 'ox' 'sheep' 'buffalo' 'pig' 'cow']
Cluster 10:
['antelope' 'horse' 'giraffe' 'zebra' 'deer']
```

then compute the cost

In [7]:
```
total_cost = 0
for i in np.unique(clusters):
    indices = np.where(clusters==i)[0]
    center = np.mean(predicates[indices], axis=0)
    cost = np.sum(np.linalg.norm(predicates[indices]-center,axis=1) ** 2)
    total_cost += cost
```

In [8]:
```
total_cost
```

Out[8]: 679136.2672672619

now for the k-means method, set it as random

In [9]:
```
cost_list=[]
clusters_list=[]
for iteration in range(10):#need to train it randomly 10 times
    k_means=sklearn.cluster.KMeans(n_clusters = 10,init='random')
    k_means.fit(predicates)
    clusters=k_means.labels_
    total_cost= 0
    each_cluster=[]
    for i in np.unique(clusters):
        indices=np.where(clusters==i)[0]
        center=np.mean(predicates[indices], axis=0)
        cost=np.sum(np.linalg.norm(predicates[indices]-center,axis=1) ** 2)
        total_cost+=cost
        each_cluster.append(('Cluster {}: '.format(i),animals[indices]))
    cost_list.append(total_cost)
    clusters_list.append(each_cluster)
```

print it out and it seems good

In [10]:
```
cost_list
```

Out[10]:
```
[741440.7291238096,
 743084.6458926192,
 739678.2472394049,
 705215.3073183551,
 728010.7960079365,
 739302.4464085281,
 717332.7650766666,
 739030.038,
 726043.1557193217,
 724722.3775195238]
```

get the lowest cost index

In [11]:
```
index_lowcost=cost_list.index(min(cost_list))
index_lowcost
```

Out[11]:
```
3
```

the lowest cost

In [12]:
```
cost_list[index_lowcost]
```

Out[12]:
```
705215.3073183551
```

the corresponding clusters

In [13]:
```
clusters_list[index_lowcost]
```

Out[13]:
```
[('Cluster 0: ', array(['grizzly+bear', 'polar+bear'], dtype='<U15')),
 ('Cluster 1: ',
  array(['antelope', 'horse', 'moose', 'giraffe', 'buffalo', 'deer'],
        dtype='<U15')),
 ('Cluster 2: ',
  array(['german+shepherd', 'tiger', 'leopard', 'fox', 'wolf', 'bobcat',
         'lion'], dtype='<U15')),
 ('Cluster 3: ',
  array(['dalmatian', 'persian+cat', 'siamese+cat', 'chihuahua', 'collie'],
        dtype='<U15')),
 ('Cluster 4: ',
  array(['ox', 'sheep', 'giant+panda', 'pig', 'cow'], dtype='<U15')),
 ('Cluster 5: ',
  array(['spider+monkey', 'gorilla', 'chimpanzee'], dtype='<U15')),
 ('Cluster 6: ', array(['zebra'], dtype='<U15')),
 ('Cluster 7: ',
  array(['beaver', 'skunk', 'mole', 'hamster', 'squirrel', 'rabbit', 'bat',
         'rat', 'weasel', 'mouse', 'raccoon'], dtype='<U15')),
 ('Cluster 8: ',
  array(['hippopotamus', 'elephant', 'rhinoceros'], dtype='<U15')),
 ('Cluster 9: ',
  array(['killer+whale', 'blue+whale', 'humpback+whale', 'seal', 'otter',
         'walrus', 'dolphin'], dtype='<U15'))]
```

In [ ]: