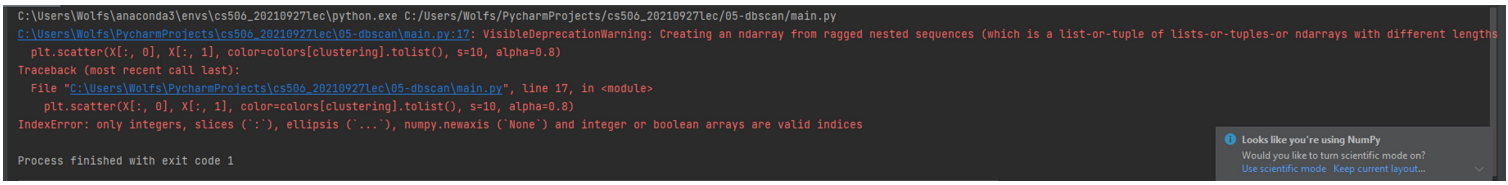
20210929

Wednesday, September 29, 2021 3:56 PM

- All of us should have received emails if you are doing a Spark project
- Team projects are difficult
  - Personalities
  - Delegate and work together
- Meet with your team regularly:
  - Even if for 5 minutes so that we can be aware of issues as fast a possible
    - HW, midterm, illnesses, etc
      - So you can budget for it
    - Be agile with what people can handle
      - If someone can't do something, then you need to adjust the scope of the project rather than just redistributing work
      - Talk with project manager (not the client) for these kinds of things
        - Ask for advice on regular basis
- Can escalate to the Prof if necessary
- If you have questions for general purpose of project
  - Talk to project manager
  - Minimize the impact on the client
  - Project manager already interfacing with client on regular basis
- HW1 is due soon
  - You should have all started
  - It is challenging

==========

DB-SCAN: note I have a bug in the following,



File - C:\Users\Wolfs\PycharmProjects\cs506_20210927lec\cs506\dbscan.py

```python
# Always want to minimize debugging space
# Try to get ahead of what he's coding and find flaws before he does
# Bredford???? search
# Top down approach


# Get distance:
from .sim import euclidean_dist

class DBC():

    def __init__(self, dataset, min_pts, epsilon):
        self.dataset = dataset
        self.min_pts = min_pts
        self.epsilon = epsilon

    def _get_epsilon_neighborhood(self, idx):
        neighborhood = []
        for i, y in enumerate(self.dataset):
            # There could be duplicate points
            # We may not want to compare a point with literally itself
            if i == idx:
                continue
            if euclidean_dist(self.dataset[idx], y) <= self.epsilon:  # Then the point is in the epsilon neighborhood of x
                neighborhood.append(i)  # Just tracking the index of the point
        return neighborhood

    def _make_bfs_assignments(self, queue, assignment, assignments):
        # We need to explore to and know if something is a core point and therefore it's neighborhood needs to be
        # explored and added to queue
        # If its a border point, it gets assignment but it doesn't get it's neighborhood added to the queue
```

```
32
33          while queue:
34
35              nextP = queue.pop(0)
36
37              if assignments[nextP] != 0:
38                  continue
39
40              assignments[nextP] = assignment  # We
    shouldn't be overwritting will go on forever otherwise
41
42              neighborhood_of_nextP = self.
    _get_epsilon_neighborhood(nextP)
43              if len(neighborhood_of_nextP) >= self.
    min_pts:
44                  # Core point
45                  queue += neighborhood_of_nextP
46
47          return assignments
48
49      def dbscan(self):
50          """
51              returns a list of assignments. The index
    of the
52              assignment should match the index of the
    data point
53              in the dataset.
54          """
55          # assignment vs assignments is not great
    naming
56          assignments = [0 for _ in range(len(self.
    dataset))]
57          assignment = 1
58          # Iterate over points in dataset,
59          # If you iterate over points themselves
60          # You need to assignment of the point to match
     the index of the point
61          # So this can get messy
62          # So iterate over indexes instead
63          # Or use enumerate to get index and the point
64
```

```
        for i, x in enumerate(self.dataset):
            if assignments[i] != 0:
                continue  # Skip the rest of this iteration of the loop
            # Need to determine whether or not X is a core point
            # This statement doesn't make sense, because you want to get the actual neighborhood from them helper:
            # if self._get_epsilon_neighborhood(x) >= self.min_pts:
            neighborhood = self._get_epsilon_neighborhood(i)
            if len(neighborhood) >= self.min_pts:
                # Core point
                # Explore x neighborhood for core points and give them the same assignment
                # Instead of doing that now, we are going to save:
                # Helper function
                assignments[i] = assignment
                assignments = self._make_bfs_assignments(neighborhood, assignments, assignments)
            assignment += 1
            # We need to account for stumbling across a core point a second time
            # Without a core point, you can't really do anything
            # Everything is initially labeled noise
            # You know it is a border point if it's in a core point's neighborhood, but isn't a core point
            # Will iterate over queue and pop off
        return assignments

# Try to do on your own before next time
# Someone please upload this code to github
# My code isn't working properly
```

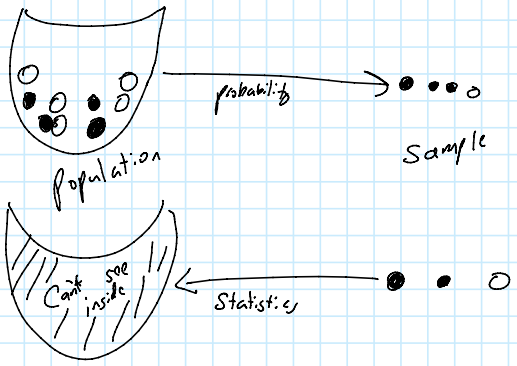
File - C:\Users\Wolfs\PycharmProjects\cs506_20210927lec\05-dbscan\main.py

```
 1 import numpy as np
 2 import matplotlib.pyplot as plt
 3 import sklearn.datasets as datasets
 4 from sklearn.preprocessing import StandardScaler
 5
 6 from cs506 import dbscan
 7
 8 centers = [[1, 1], [-1, -1], [1, -1]]
 9 X, _ = datasets.make_blobs(n_samples=750, centers=
   centers, cluster_std=0.4,
10                            random_state=0)
11 plt.scatter(X[:,0],X[:,1],s=10, alpha=0.8)
12 plt.show()
13
14 clustering = dbscan.DBC(X, 3, .2).dbscan()
15 colors = np.array([x for x in '
   bgrcmykbgrcmykbgrcmykbgrcmyk'])
16 colors = np.hstack([colors] * 20)
17 plt.scatter(X[:, 0], X[:, 1], color=colors[clustering
   ].tolist(), s=10, alpha=0.8)
18 plt.show()
19
```

20210929 Probability

Wednesday, September 29, 2021 4:59 PM

Probability vs. Statistics → reasoning in opposite directions



Example: Coin Toss

$\Omega = \{H, T\}$

Omega: set of all possible outcomes

script f: all possible events

$\mathcal{F} = \{\emptyset, \{H\}, \{T\}, \{H, T\}\}$

$P: \mathcal{F} \to [0, 1]$

$(\Omega, \mathcal{F}, P)$ → a probability space

where

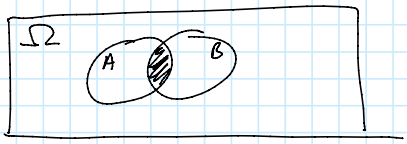
- $P(\Omega) = 1$ (1st axiom)
- $P(A) \geq 0, \quad A \in \mathcal{F}$

  any possible outcome prob. is @ lst 0
- $P\left(\bigcup_{i=1} A_i\right) = \sum_{j=1}^{n} P(A_j)$

  if $A_i \cap A_j = \emptyset$ ← assuming not

  double check subscripts!
- $P(A^c) = 1 - P(A)$
- $P(A) \leq 1 \to P(A^c) \geq 0$
- $P(\emptyset) = 0$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

  ↑ subtract intersection



- $P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \mid A) P(A)}{P(B)}$

  ← Bayes Rule

  given

- A & B are independent if:

  $P(A \cap B) = P(A) \cdot P(B)$

$$\begin{cases} P(A \cap B) = P(A) \cdot P(B) \\ P(A \mid B) = P(A) \\ P(B \mid A) = P(B) \end{cases}$$

Focus back on coin toss

Distribution characterizes the population

- If you list all possible outcomes & their probabilities, then you've characterized the population

Use Random Variable

A random variable $X: \Omega \to \mathbb{R}$
maps outcomes to real line

lowercase omega

$$X(\omega) = \begin{cases} 1 & \text{if } \omega = \{H\} \\ 0 & \text{if } \omega = \{T\} \end{cases}$$

$$P_X(1) = P(X = 1)$$

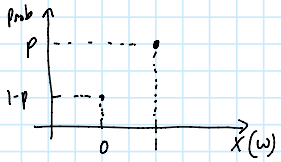
Using $P: \mathcal{F} \to [0, 1]$

$$= P(\{\omega \mid X(\omega) = 1\})$$
$$= P(\{H\})$$

Extremely rigorous way of describing mapping

What is the distribution of our random var X

Distribution of our coin toss



May be unfair coin

Bernoulli Trial

$X_1, X_2 \ldots$
X_1: 1st coin flip, X_2: 2nd

What is the probability that it will take k # of tosses to get first heads?

Probability that it takes k tosses to get first H.

$$P_{X_1}(0)\, P_{X_2}(0)\, P_{X_3}(0) \ldots P_{X_k}(1)$$

assuming independence of random variables

$$= (1-p)^{k-1} \cdot p$$

assuming independence of random variables

$= (1-p)^{k-1} \cdot p$

Geometric Distribution

What probability that you get k # of heads in first n coin flips?

Prob of getting k Hs in n tosses?

Binomial Distribution

$$P_n(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

n choose k

On average, you are getting ~60 heads per hour.

Flip coin over large period of time, on average we get 60 Hs/hour.

Probability of k heads in a minute?

↳ need a different distribution function

Poisson Dist

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

λ rate $= 1 = \mu$ ← down to our unit

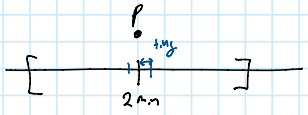
double check

Maximum when $k = 1$?

$\mu = 1$/min

What is probability that b/w 2 successes is exactly 2 min?

↳ ∞ # of values that are very close to 2 min value



No way to sum ∞ # of values & have it equal one

↳ So this ? doesn't make sense for probability

B/w 1min & 2min makes more sense in probability

Time domain is continuous, but we have been talking about discrete outcomes.

Can't list things w/ continuum of values.

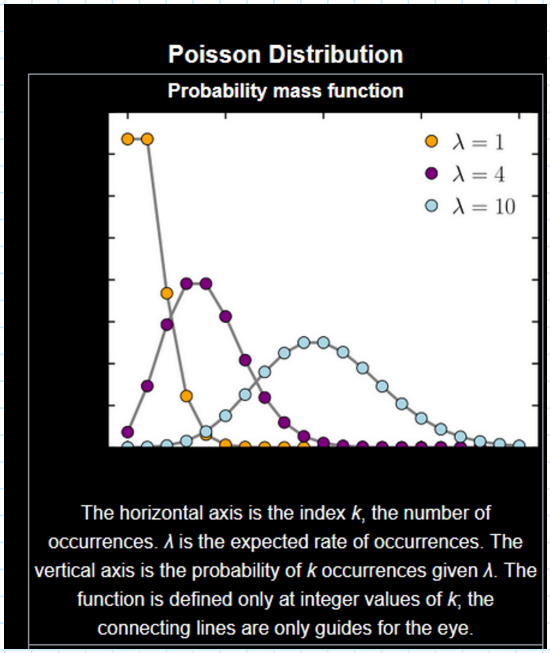
With a continuum of possible outcomes, we want to look @

$P(X < \epsilon)$ ← Cumulative Distribution

Exponential Distribution

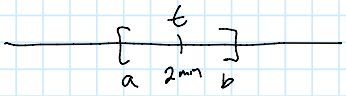
P(time b/w consecutive Hs) $= e^{-\mu t}$

https://en.wikipedia.org/wiki/Poisson_distribution



Instantaneous Prob. Dist.

Instantaneous change of prob @ point



as $\Delta t \to 0$, what happens in this infantesimally small range

$$f(t) = \mu e^{-\mu t}$$ (is a derivative)

Probability Density Function

$$P(t \in [t_1, t_2]) = \int_{t_1}^{t_2} \mu e^{-\mu t} dt = P(t \geq t_1, t \leq t_2)$$

If you were to average the time b/w successive heads?.

↳ What is this probability distribution

↳ normal distribution

↳ Central limit theorem

All these different distributions can be covered just be flipping a coin