DS2500

1/24 - Fri !!

Admin

- HW1 due 9pm
- HW2 out, due 1/31 9pm
- Lab 2 monday 1/27

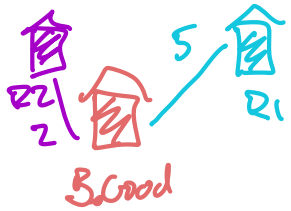
Agenda

1. Similarity + Distance
2. Qualitative Distance
3. Python

- DS2500 exams
  - ↳ 2/14, 4/4
  - in class, on paper
  - entire lecture to complete
  - 8.5x11 cheat sheet (one side)
  - practice out week of 2/3
  - DAS, schedule time

1. Similarity + Distance

↳ distance: math pov
numeric value



Similarity: DS pov
apply a distance formula to quantify similarity

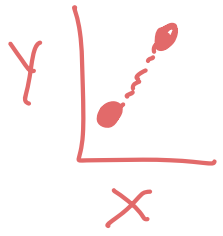
using similarity in DS:
↳ how similar are two... movies? Recommendation
diseases? vaccine protocol
words? spelling correction
images? Facial recognition

Data
- quant — discr / cont. } distance ☑ (HW2)
- qual — nom * distance? ☑ yes, kind of (today)
  - ord — nah

Quantitative: compute distance between 2 objects
↳ we look at compatible features
↳ same category
same scale
↳ like an attribute
Food: prox, rating, wait, etc.

Default: Euclidean Distance (ratio)
Hamming Distance (nominal)

Euclidean Distance ↳ two objects, p and q
p, q have same features, are numeric



$$\delta(p,q) = \sqrt{\sum (q_i - p_i)^2}$$

$\sum$ sum
q_i, p_i features

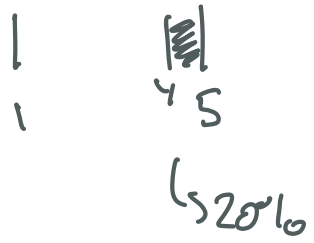
(ex)

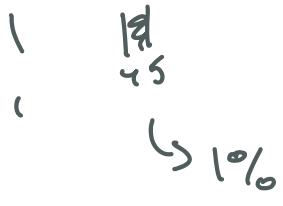
	prox	rating	wait
P	4	5	2
Q	1	4	5

$$\delta(P,Q) = \sqrt{(1-4)^2 + (4-5)^2 + (5-2)^2}$$
$$= \sqrt{-3^2 + -1^2 + 3^2}$$
$$= \sqrt{9+1+9}$$
$$= \sqrt{19}$$
$$= 4.36$$

* Food features (qual) are 1-5

Euclidean assumes unit change in any feature is equally significant





3. Qualitative Distance

↳ Hamming distance: count # features that differ

(ex) Spelling/strings

count # positions w/different characters

northeastern

northwestern

dist = 2

What works for Food objects

- name
- description
- hours

→ B. Good ? !! El Sete ?

Indoor seating, vegan-friendly, good for takeout
Good system/ quick takeout, great indoor atmosphere
Indoor Seating, Good for Takeout
Indoor seating, good for takeout, vegan friendly, has a bathroom
Good for takeout, quick service

- indoor 0/1
- vegan 0/1
- bathroom 0/1

yes/no labels
nominal, not numeric

(ex)

	indoor	vg	bath
R1	1	0	1
R2	0	1	1

Hamming — # bits flipped

d(R1, R2) = 2

(ex) Hamming w/numeric (treated as nominal)

P	4	5	2
Q	3	2	4

Ham(P, Q) = 3

(less nuanced less info than Euclidean)

3. Python

- Build an Food class from Yves
  - attributes: name, rating, etc.

Adding on:
- read from csv file (as2500-gen data!)
- create Food objects
- turn description into 0/1
- Hamming distance method

Goal:
↳ create same objects

Compute distance to B. Good

```
Indoor seating, vegan-friendly, good for takeout
Good system/ quick takeout, great indoor atmosphere
Indoor Seating, Good for Takeout
Indoor seating, good for takeout, vegan friendly, has a bathroom
Good for takeout, quick service
```

- vegan 0/1
- takeout 0/1
- quick 0/1

↳ Food attributes

```
self.vg = 0/1
self.take = 0/1
self.quick = 0/1
```

↳ Dict attribute

```
{'vg': 0/1,
 'take': 0/1,
 'quick': 0/1}
```

Diner:
- iterate over same rows of 2D list
- each row → Food object
- in the row, description at end of row

↳ row[-1]

- look for keywords in description
  ↳ "vegan" in descr → 0/1

"take-out" in descr -0/1

self, other are objects

vegan
takeout → attributes
quick

```
def ham_dist(self, other):
    dist = |self.vg - other.vg| + |self.takeout - other.takeout| + |self.quick
                                                                   - other.quick|
```

0 0 } 0
1 1 }

1 0 } 1
0 1 }

```
    return dist
```