Discussion Week 7

Agenda

- Statistics Introduction/Review
  - Random Variables
  - Probability
  - Conditional Probability
  - Bayes Rule
- Vector embedding distances/similarity measures
  - Euclidean, Manhattan, Cosine
- HW3
  - Any questions?

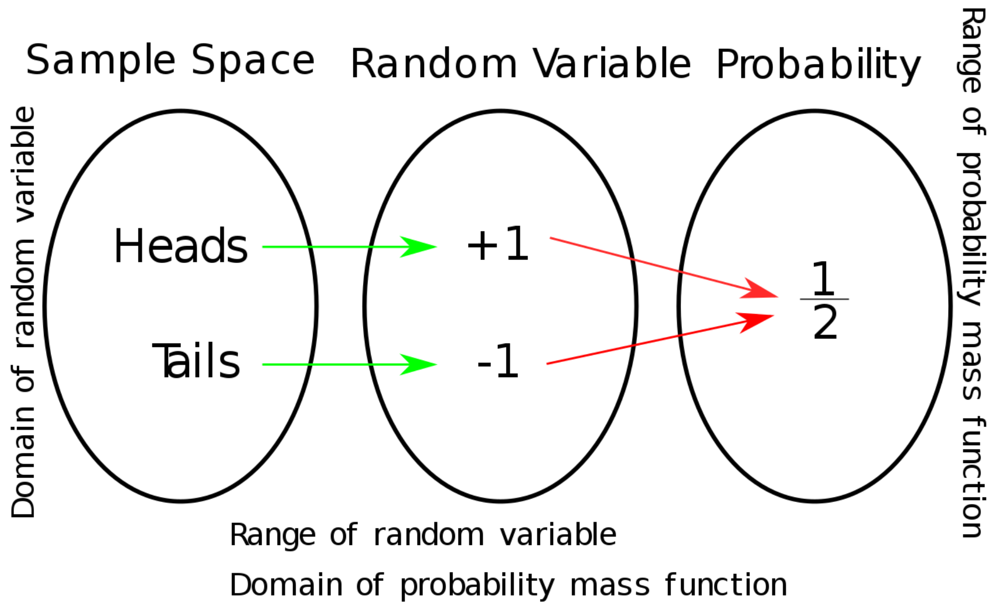
HW Submission and Other Reminders

- Please make sure that you select your pages when submitting the HW
  - There are ~150 students in the course and it makes grading difficult if we need to scan your PDF to find your answers
- Make sure you submit the PDF and .ipynb files into their respective submissions on Gradescope
  - We sometimes use the .ipynb files if there are large issues with your submission
  - If you still have troubles with this, please let me know and I would be happy to help you
- Congrats on completing the midterm!
  - **One piece of advice**: While showing each step of your calculations (such as eigenvalue calculation, etc), the work you show or explain, the more partial credit we can give as graders
  - If you don't show any work and give an incorrect answer, there is no way for us to see where you went wrong and give any partial credit!

Probability

Random Variables

- A random variable is a variable whose possible values are numerical outcomes of a random phenomenon
- A random variable X has a domain of possible values
  - For a discrete variable, it will have a domain of the form $\{x_1, x_2, \dots, x_m\}$
  - A continuous random variable could take on an infinite number of possible values.
- Discrete random variable example
  - Coin flip C with possible values
    - $\{c_1 = Heads, c_2 = Tails\}$



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

Unconditional Probability (prior)

- The unconditional probability $\mathrm{P}(\mathrm{X} = x_i)$ denotes our degree of belief that $\mathrm{X} = x_i$ in the absence of all other knowledge

- e.g., What is the probability that we flip a heads? Flip a tails?
  - $\mathrm{P}(\mathrm{C} = \mathrm{Heads}) = 0.5$
  - $\mathrm{P}(\mathrm{C} = Tails) = 0.5$ **or** P(C=Tails) = $1 \; - \mathrm{P}(\mathrm{C} = \mathrm{Heads})$

Axioms of Probability

- Probabilities are non-negative
  - $\mathrm{P}(\mathrm{X} = x_i) \geq 0$
- Probabilities are normalized or they sum to one:
  - $\sum_i P(X = x_i) = 1$
- Probabilities add for the union of mutually exclusive events
  - $P\left(X = x_i \text{ or } X = x_j\right) = \mathrm{P}(\mathrm{X} = x_i) + P\left(X = x_j\right)$
    - If $x_i \neq x_j$

Conditional Probability

- The conditional probability $\mathrm{P}(\mathrm{X} = x_i | Y = y_j)$ denotes our belief that $\mathrm{X} = x_i$ given $Y = y_j$.

- Example: Drawing cards
  - Probability that the suit of the card is hearts
    - $\mathrm{P}(\mathrm{S} = \mathrm{Hearts}) = \frac{13}{52} = 0.25$
  - Probability that the color of the card is red
    - $\mathrm{P}(\mathrm{C} = Red) = \frac{26}{52} = 0.5$
  - Probability that we draw a hearts given we know the color
    - $\mathrm{P}(\mathrm{S} = \mathrm{Hearts}|\mathrm{C} = \mathrm{Red}) = \frac{13}{26} = 0.5$ *(higher)*
    - $\mathrm{P}(\mathrm{S} = \mathrm{Hearts}|\mathrm{C} = Black) = \frac{0}{26} = 0$ *(lower)*

Axioms of Probability Apply to Conditional Probability!

- Conditional probabilities are also non-negative
  - $\mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) \geq 0$

- Conditional probabilities are normalized, or they sum to one:
  - $\sum_i \mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) = 1$

- Probabilities add for the union of mutually exclusive events
  - $P\left(X = x_i \text{ or } X = x_j | Y = y_k\right) = \mathrm{P}(\mathrm{X} = x_i | Y = y_k) + P\left(X = x_j | Y = y_k\right)$
    - If $x_i \neq x_j$

- Notice, these axioms are the same!

Conditional Probability Check:

- We've seen that conditional probabilities sum to 1 when summed over the variables in the left side of the conditioning bar
  - $\sum_i \mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) = 1$
- However, what about if we sum over the variable that we are conditioning on?
  - **True or False:** Does the following statement always hold?
    - $\sum_j \mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) = 1?$

<u>Conditional Probability Check:</u>

- We've seen that conditional probabilities sum to 1 when summed over the variables in the left side of the conditioning bar
  - $\sum_i \mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) = 1$

- However, what about if we sum over the variables we are conditioning on?
  - **True or False:** Does the following statement always hold?
    - $\sum_j \mathrm{P}\left(\mathrm{X} = x_i | Y = y_j\right) = 1?$
  - <u>**False. This does not necessarily sum up to 1!**</u>

More Rules

- The Product Rule:
  - $\mathrm{P}(\mathrm{X} = x_i, Y = y_j) = \mathrm{P}(\mathrm{X} = x_i) P(Y = y_j | X = x_i)$
    - Similarly
  - $\mathrm{P}(\mathrm{X} = x_i, Y = y_j) = P(Y = y_j) P(X = x_i | Y = y_j)$

- Marginalization:
  - $\mathrm{P}(\mathrm{X} = x_i) = \sum_j P(X = x_i, Y = y_j)$

- Conditional Probability:
  - $P(X = x_i | Y = y_j) = \frac{P(X = x_i, Y = y_j)}{P(Y = y_j)}$

- Bayes Rule:
  - $P(X = x_i | Y = y_j) = \frac{\mathrm{P}(Y = y_j | X = x_i) P(X = x_i)}{P(Y = y_j)}$

Bayes Rule

- So, when do we want to use Bayes rule?
  - We can utilize Bayes rule when we might need to express $P(X = x_i | Y = y_j)$ in terms of things that <u>we already know</u> or would be <u>easier to calculate</u>.

$$\underbrace{\boxed{P(A|B)}}_{\text{posterior}} = \underbrace{\boxed{P(A)}}_{\text{prior}} \times \frac{\boxed{P(B|A)}\ \text{likelihood}}{\boxed{P(B)}\ \text{marginal}}$$

https://www.freecodecamp.org/news/bayes-rule-explained/

Bayes Rule from Conditional Probability Formula

- From the definition of conditional probability on the slide previous:

$$P(X = x_i | Y = y_j) = \frac{\mathrm{P}(Y = y_j | X = x_i)P(X = x_i)}{P(Y=y_j)}$$

Starting from the LHS:

$\Rightarrow \frac{P(X=x_i,Y=y_j)}{P(Y=y_j)}$ (Conditional Probability)

$\Rightarrow \frac{P(X=x_i)P(Y=y_j|X=x_i)}{P(Y=y_j)}$ (product rule)

Short Example:

- For a 6-sided dice, what's the probability that the outcome is at least 4, given that it is even?
  - Valid outcomes: 4 and 6

- Probability that the dice is at least 4?
  - $P(D \geq 4) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

- Probability that the dice is even?
  - $P(D \in even) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

- Probability that the dice is even and at least 4?
  - $P(D \in even, D \geq 4) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$

Using the formula for conditional probability:

$$P(D \geq 4 |\, D \in even) = \frac{P(D \in even, D \geq 4)}{P(D \in even)} \Rightarrow \frac{2/6}{3/6} = \frac{2}{3}$$

Using Bayes rule:

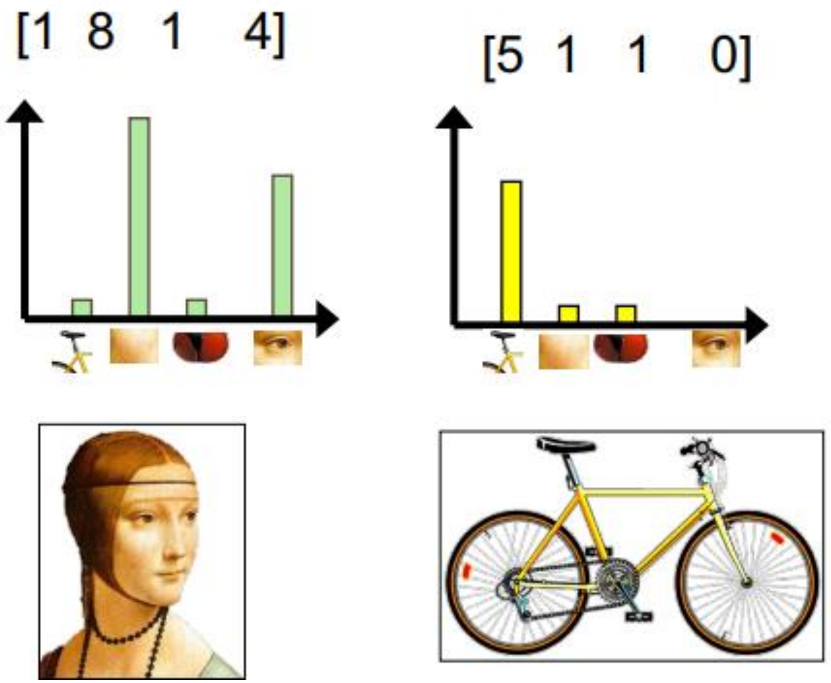
$$P(D \geq 4 |\, D \in even) =$$

$$\frac{P(D \geq 4)\mathrm{P}(D \in even | D \geq 4)}{P(D \in even)} \Rightarrow \frac{\frac{3}{6} * \frac{2}{3}}{\frac{3}{6}} = \frac{2}{3}$$

Vector Distance/Similarity Measures

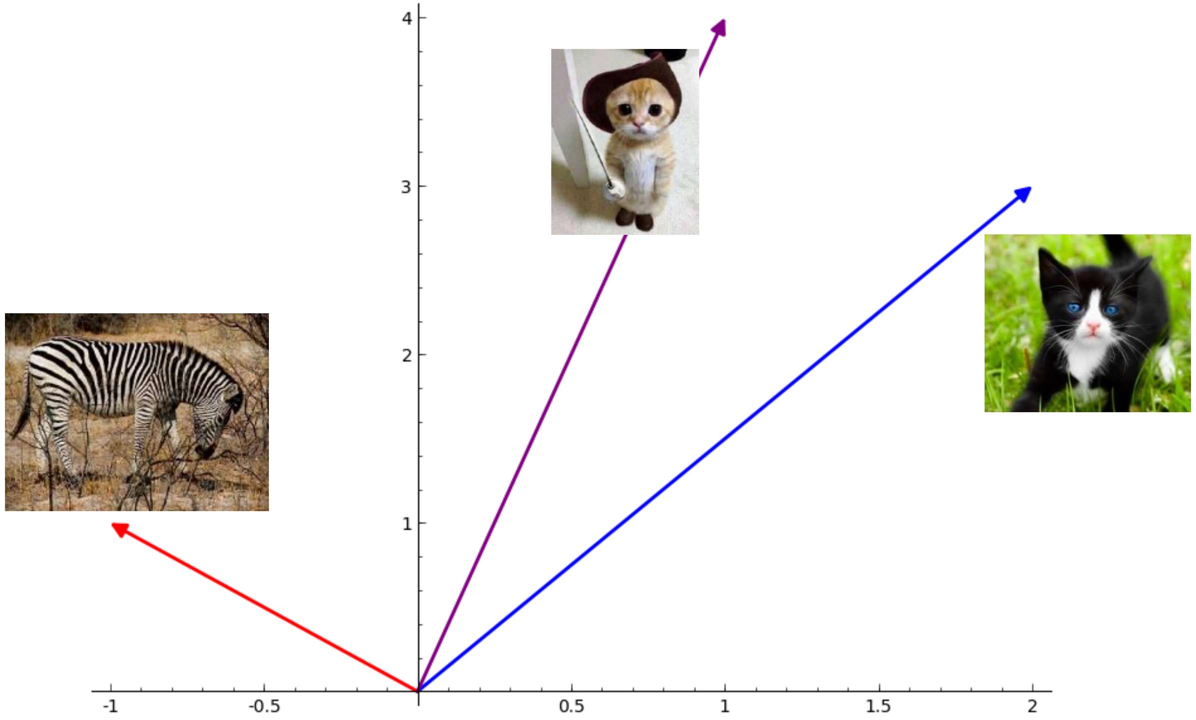
HW3 Question 1: Creating Vector Representations of Images using Visual Bag of Words

- In HW3, we create vector representations of our images using a histogram of our visual vocabulary



Plotting the Histograms of the Images as Vectors

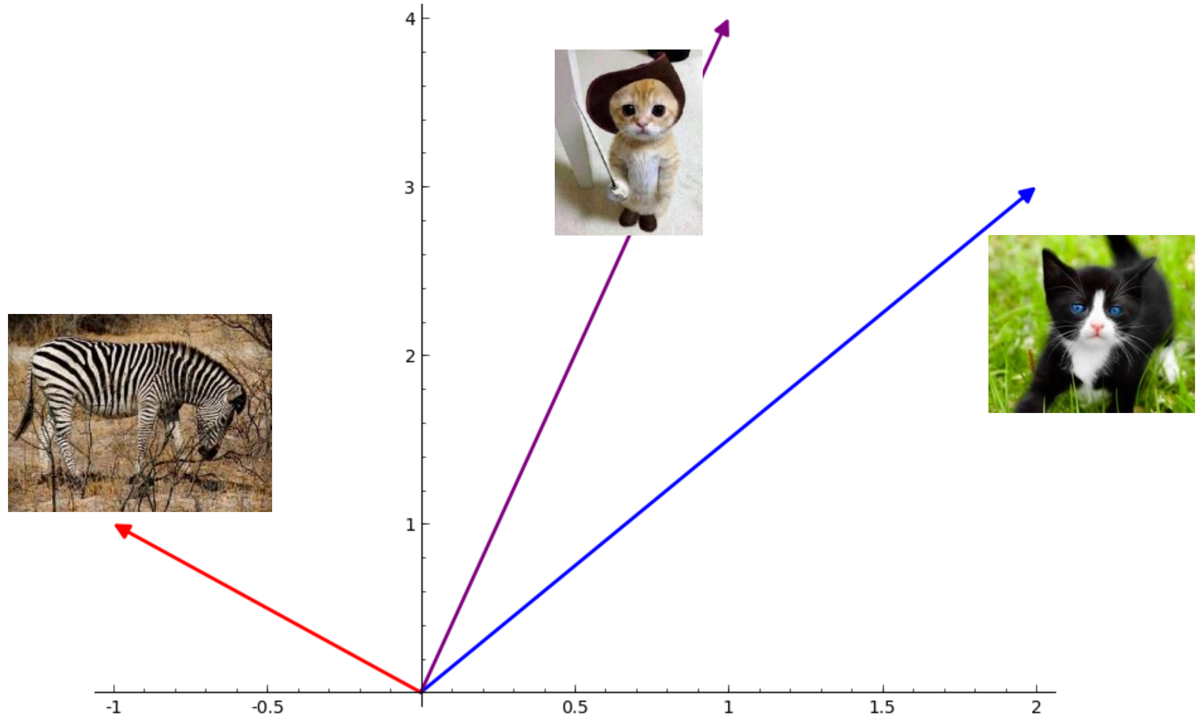
- Once we have created the histograms of each of our images, we can treat the histograms as vectors (likely higher dimension than 2D or 3D).
- To simplify the visualization, imagine the histograms of the images plotted as 2D vectors like so:



https://genedan.com/tag/vector-plot-sage/

Plotting the Histograms of the Images as Vectors

- Ideally, images that display similar objects would be "closer" to each other in our vector space than others
- Here, we can see that the vectors of the two cats are "closer" or "more similar" to each other than to the zebra.

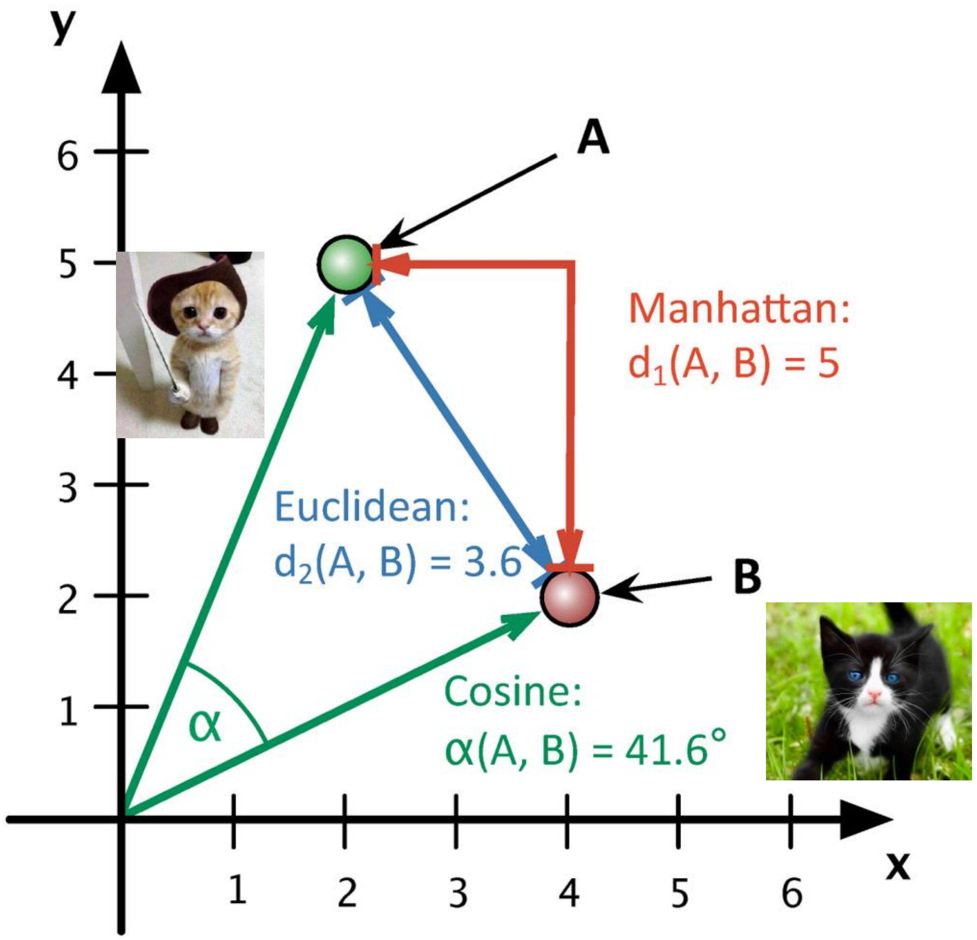


https://genedan.com/tag/vector-plot-sage/

Vector Distance Measures

- However, we want to quantify the “closeness” or similarity between the vectors
- The most obvious measure is the Euclidean distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

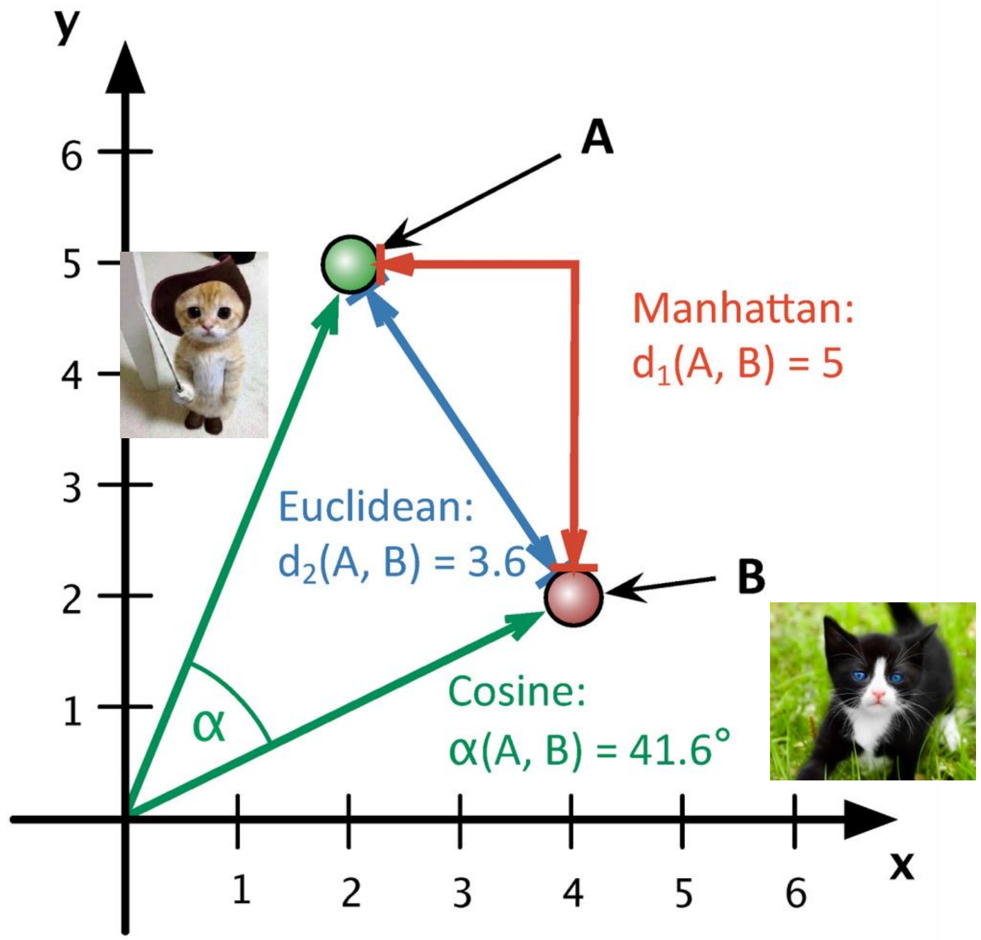


https://medium.com/@prasoonthakur5/different-types-of-distances-used-in-machine-learning-7491128491b8

Vector Distance Measures

- But there are others! As mentioned in lecture, we talked about the cosine similarity
- Here, we use the cosine of the angle between the vectors
  - Vector length here makes no difference!

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

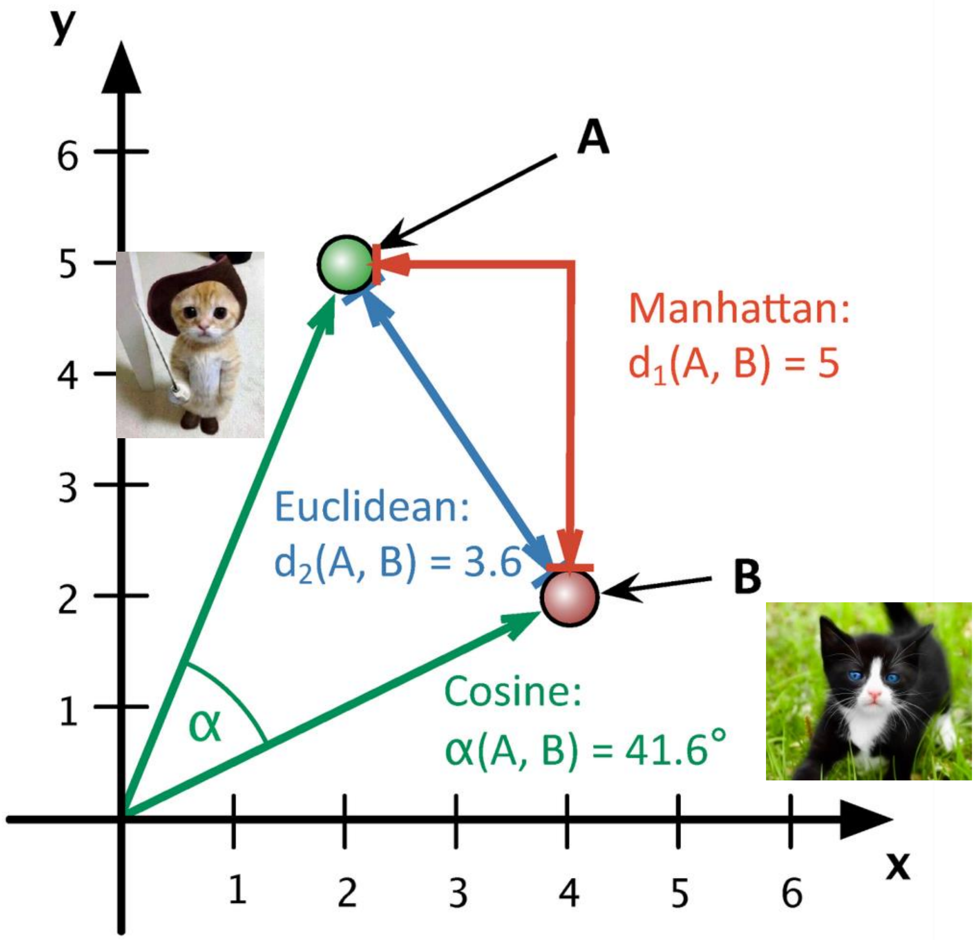


https://medium.com/@prasoonthakur5/different-types-of-distances-used-in-machine-learning-7491128491b8

Vector Distance Measures

- Another one is the Manhattan distance
  - Think walking around the streets of Manhattan!

$$\sum_{i=1}^{n} |p_i - q_i|$$



https://medium.com/@prasoonthakur5/different-types-of-distances-used-in-machine-learning-7491128491b8

Vector Distance Measures

- There are infinitely many more measures we can use
  - Thinking back to discussion 1, both the Manhattan distance and the Euclidean distance are based on the L-p norm.

- Euclidean distance -> L-2 Norm
- Manhattan distance -> L-1 Norm

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- The default distance used in Scipy’s KNN is Euclidean distance
  - This is fine to use!

HW3 Question 1.6

- Main idea for Q1.6:

1. Choose some combination of hyperparameters (e.g. uniform sampling, patch features, K=3)
2. Build a visual vocabulary from the <u>entire training set</u>
3. Populate the KNN with the histograms from the <u>entire training set</u>
4. Calculate accuracy on the <u>test sets:</u>
    1. Classify the histograms from the images in the <u>positive test set</u>
    2. Calculate accuracy on the <u>positive test set</u>
    3. Classify the histograms from the images in the <u>negative test set</u>
    4. Calculate the accuracy on the <u>negative test set</u>
5. Repeat for the rest of the combinations: the methods for interest points, feature descriptors, and number of neighbors in KNN

<u>For each combination, you should only create 1 visual vocabulary!</u>

- **<u>One way to put this into words is that we want our classifier to use the "same vocabulary" during training time and test time</u>**
- You should however make a new vocabulary and KNN once you complete calculating accuracy on both test sets and changing to a different combination of hyperparameters.

HW3 Question 1 Debugging Help

- Read the descriptions of the functions to determine how you should format the variables before returning them
- **<u>For debugging only</u>**, I would recommend shrinking the number of images in the training set. You can use this to make the test code run faster to debug basic syntax and runtime errors
  - Please make sure you set the reset back to the entire training set before running 1.6 and submitting!
- Accuracy for this assignment is poor. You can expect 40-60% accuracy on each test set with the implementation choices we’ve made. You will not be marked down for this.
  - Perhaps we can increase the performance with different implementation choices