Data Mining & Machine Learning

CS37300
Purdue University

Sep 8, 2023

Your First Classifier!

- Let's consider one of the simplest classifiers out there.
- Assume we have a training set $(x_1,y_1)\ldots(x_n,y_n)$
- Now we get a new instance x_{new} , how can we classify it?
  - Example: Can you recommend a movie, based on user's movie reviews?

Your First Classifier!

- Let's consider one of the simplest classifiers out there.
- Assume we have a training set $(x_1,y_1)\ldots(x_n,y_n)$
- Now we get a new instance x_{new} , how can we classify it?
    - Example: Can you recommend a movie, based on user's movie reviews?
- **Simple Solution**:
    - Find the most similar example (x,y) in the training data and predict the same
        - If you liked "*Fast and Furious*" you'll like "*2 fast 2 furious*"
- Only a single decision is needed: distance metric to compute similarity

Your First Classifier!

- Let's consider one of the simplest classifiers out there.
- Assume we have a training set $(x_1,y_1)\ldots(x_n,y_n)$
- Now we get a new instance x_{new} , how can we classify it?
  - Example: Can you recommend a movie, based on user's movie reviews?
- **Simple Solution**:
  - Find the most similar example (x,y) in the training data and predict the same
    - If you liked "*Fast and Furious*" you'll like "*2 fast 2 furious*"
- Only a single decision is needed: distance metric to compute similarity

$$d(x_1, x_2) = 1 - \frac{x_1 \bigcap x_2}{x_1 \bigcup x_2} \qquad d(x_1, x_2) = \sqrt[2]{(x_1 - x_2)^2}$$

K Nearest Neighbors

- Can you think about a better way?

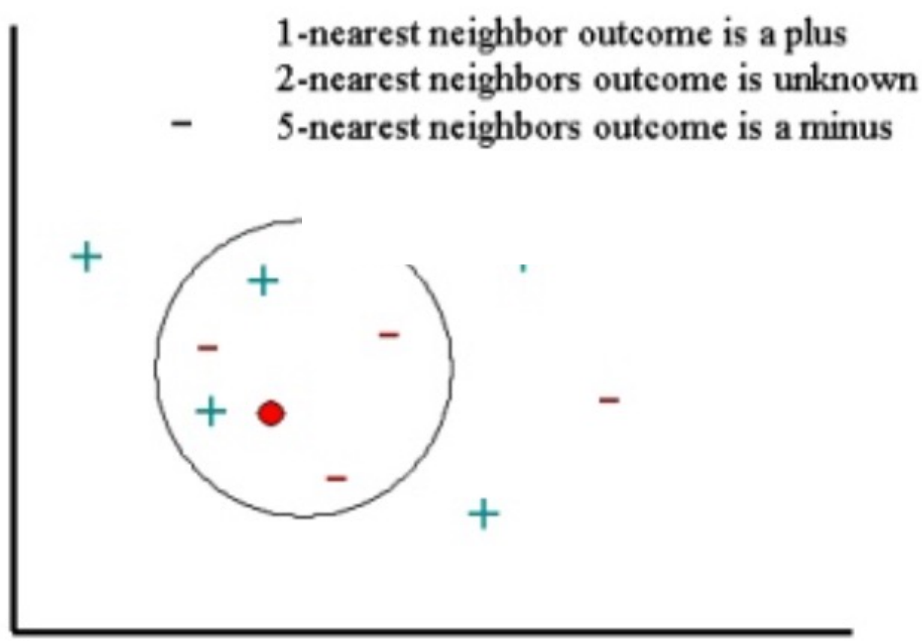
K Nearest Neighbors

- Can you think about a better way?

- We can make the decision by looking at several near examples, not just one. *Why would it be better?*

K Nearest Neighbors

- Can you think about a better way?
- We can make the decision by looking at several near examples, not just one. *Why would it be better?*



K Nearest Neighbors

- **Learning**: just storing the training examples
- **Prediction**:
  - Find the K training example closest to **x**
- **Predict a label:**
  - Classification: majority vote
  - Regression: mean value

K Nearest Neighbors

- **Learning**: just storing the training examples
- **Prediction**:
  - Find the K training example closest to **x**
- **Predict a label:**
  - Classification: majority vote
  - Regression: mean value
- KNN is a type of *instance based learning*
- This is called *lazy* learning, since most of the computation is done at prediction time

Let's analyze KNN

- ***What are the advantages and disadvantages of KNN?***
  - *What should we care about when answering this question?*
- ***Complexity***
  - ***Space** (how memory efficient is the algorithm?)*
    - *Why should we care?*
  - ***Time** (computational complexity)*
    - *Both at training time and at test (prediction) time*
- ***Expressivity***
  - *What kind of functions can we learn?*

Let's analyze KNN

- ***What are the advantages and disadvantages of KNN?***
  - *What should we care about when answering this question?*
- ***Complexity***
  - ***Space*** *(how memory efficient is the algorithm?)*
    - *Why should we care?*
  - ***Time*** *(computational complexity)*
    - *Both at training time and at test (predict*
- ***Expressivity***
  - *What kind of functions can we learn?*

Training is very fast! But *prediction is slow*
- O(dN) for N examples with d attributes
- *increases* with the number of examples!

Let's analyze KNN

- ***What are the advantages and disadvantages of KNN?***
  - *What should we care about when answering this question?*
- ***Complexity***
  - ***Space*** *(how memory efficient is the algorithm?)*
    - *Why should we care?*

KNN needs to maintain all training examples!
-Datasets can be HUGE

  - ***Time*** *(computational complexity)*
    - *Both at training time and at test (predict*

Training is very fast! But *prediction is slow*
- O(dN) for N examples with d attributes
- *increases* with the number of examples!

- ***Expressivity***
  - *What kind of functions can we learn?*

Analyzing K Nearest Neighbors

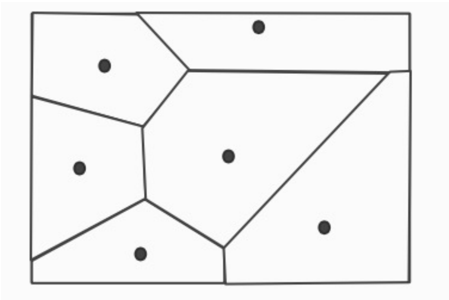
- We discussed the importance of finding a good model space
  - Expressive (we can represent the right model)
  - Constrained (we can search effectively, using the data we have)
- Let's try to characterize the model space, by looking at the **decision boundary**

Analyzing K Nearest Neighbors

- We discussed the importance of finding a good model space
    - Expressive (we can represent the right model)
    - Constrained (we can search effectively, using the data we have)
- Let’s try to characterize the model space, by looking at the **decision boundary**
- **How would it look if K=1?**

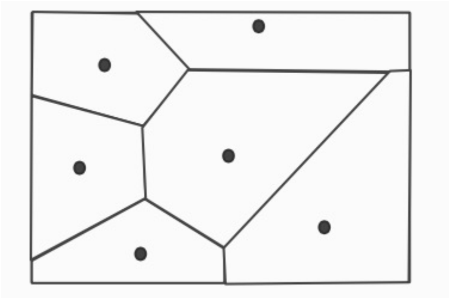
Analyzing K Nearest Neighbors

- We discussed the importance of finding a good model space
  - Expressive (we can represent the right model)
  - Constrained (we can search effectively, using the data we have)
- Let's try to characterize the model space, by looking at the **decision boundary**
- **How would it look if K=1?**



Analyzing K Nearest Neighbors

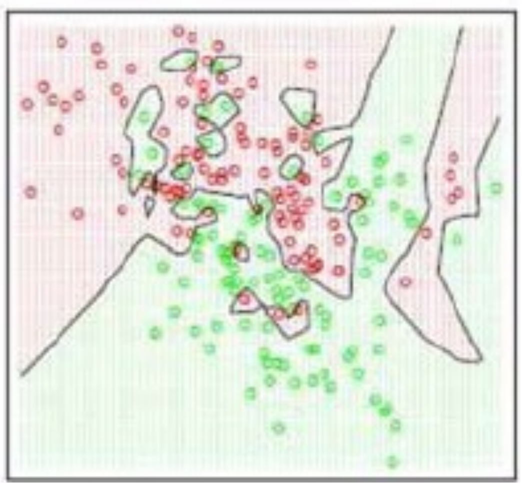
- We discussed the importance of finding a good model space
  - Expressive (we can represent the right model)
  - Constrained (we can search effectively, using the data we have)
- Let's try to characterize the model space, by looking at the **decision boundary**
- **How would it look if K=1?**

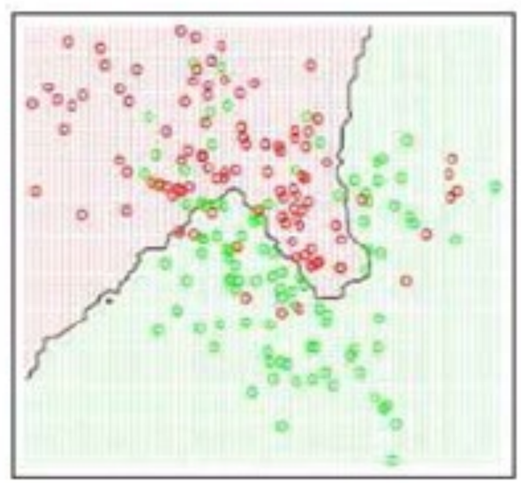


If we define the model space to be our choice of K
Does the complexity of the model space increase of decrease with K?

Analyzing K Nearest Neighbors

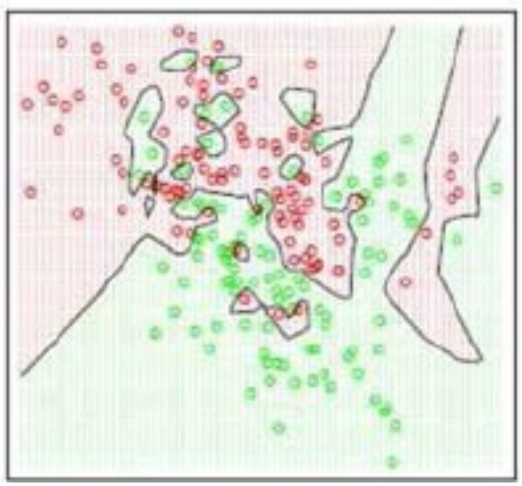
- Which model has a higher K value?

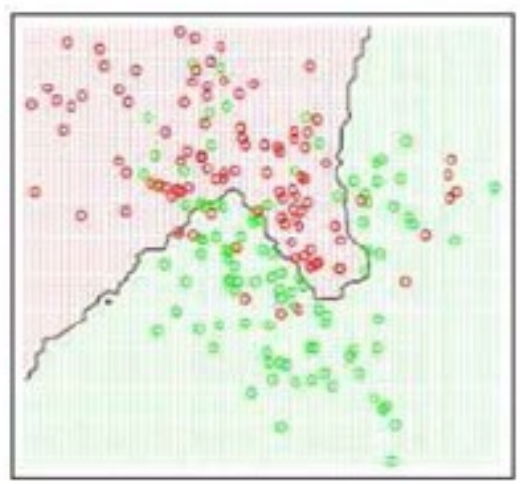




Analyzing K Nearest Neighbors

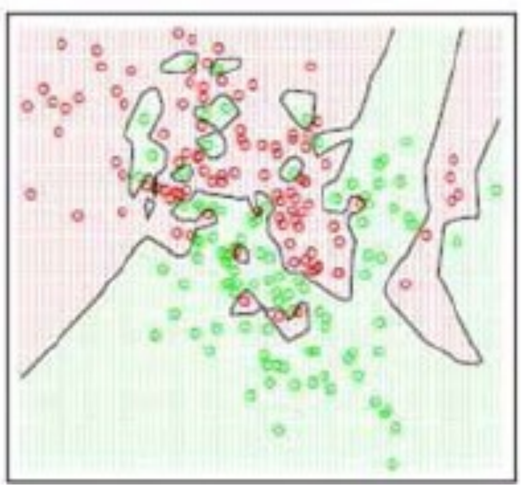
- Which model has a higher K value?
- Which model is more complex?

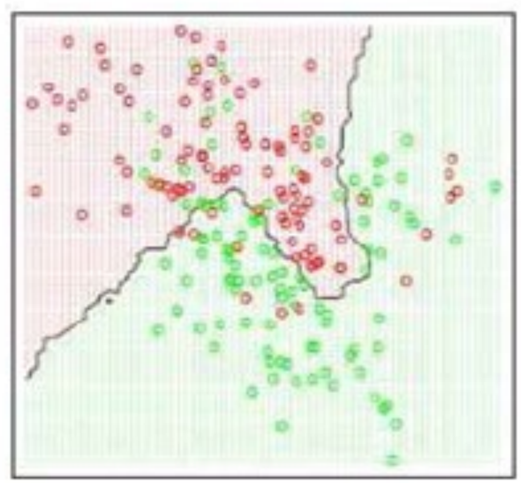




Analyzing K Nearest Neighbors

- Which model has a higher K value?
- Which model is more complex?
- Which model is more sensitive to noise?





Questions

- We know higher K values result in a smoother decision boundary.
  - Less ”jagged” decision regions
  - Total number of regions will be smaller

Questions

- We know higher K values result in a smoother decision boundary.
  - Less ”jagged” decision regions
  - Total number of regions will be smaller

What will happen if we keep increasing K, up to the point that K=n ?

n = is the number of examples we have

How should we determine the value of K?

- Higher K values result in less complex functions (less expressive)
- Lower K values are more complex (more expressive)
- **How can we find the right balance between the two?**

How should we determine the value of K?

- Higher K values result in less complex functions (less expressive)
- Lower K values are more complex (more expressive)
- **How can we find the right balance between the two?**
- Option 1: Find the K that minimizes the training error.
  - <u>Training error</u>: after learning the classifier, what is the number of errors we get on the training data.
  - What will be this value for k=1, k=n, k=n/2?

How should we determine the value of K?

- Higher K values result in less complex functions (less expressive)
- Lower K values are more complex (more expressive)
- **How can we find the right balance between the two?**
- Option 1: Find the K that minimizes the training error.
  - Training error: after learning the classifier, what is the number of errors we get on the training data.
  - What will be this value for k=1, k=n, k=n/2?

Is this a good idea?

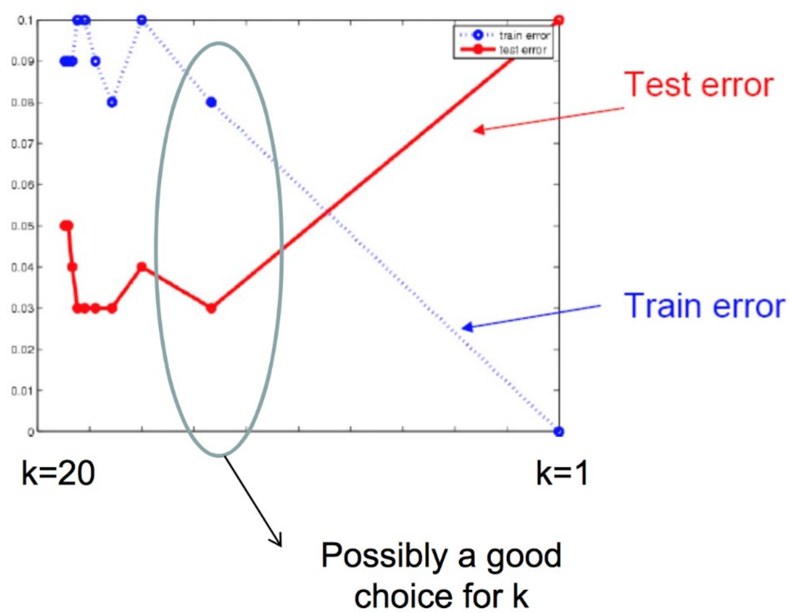
How should we determine the value of K?

- Higher K values result in less complex functions (less expressive)
- Lower K values are more complex (more expressive)
- **How can we find the right balance between the two?**
- Option 1: Find the K that minimizes the training error.
  - Training error: after learning the classifier, what is the number of errors we get on the training data.
  - What will be this value for k=1, k=n, k=n/2?

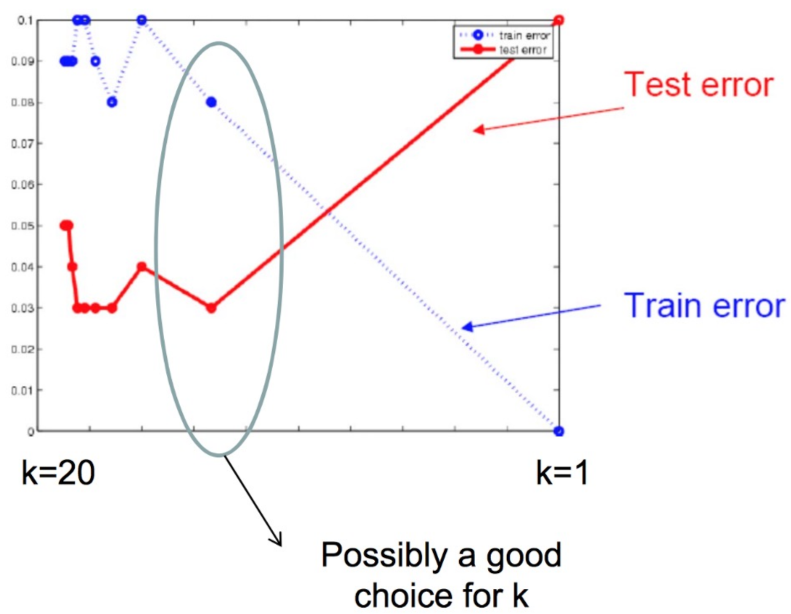
Is this a good idea?

- Option 2: Find K that minimizes the **validation error**.
  - Validation error: set aside some of the data (validation set). what is the number of errors we get on the validation data, after training the classifier.

How should we determine the value of K?



How should we determine the value of K?



In general – using the training error to tune parameters will always result in a more complex hypothesis! **(why?)**

KNN Practical Consideration

- Finding the right representation is key
  - KNN is very sensitive to irrelevant attributes
- Choosing the right distance metric is important
  - Many options!
  - Popular choices:

– Euclidean distance

$$||\mathbf{x}_1 - \mathbf{x}_2||_2 = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_{1,i} - \mathbf{x}_{2,i})^2}$$

– Manhattan distance

$$||\mathbf{x}_1 - \mathbf{x}_2||_1 = \sum_{i=1}^{n} |\mathbf{x}_{1,i} - \mathbf{x}_{2,i}|$$

– L_p-norm

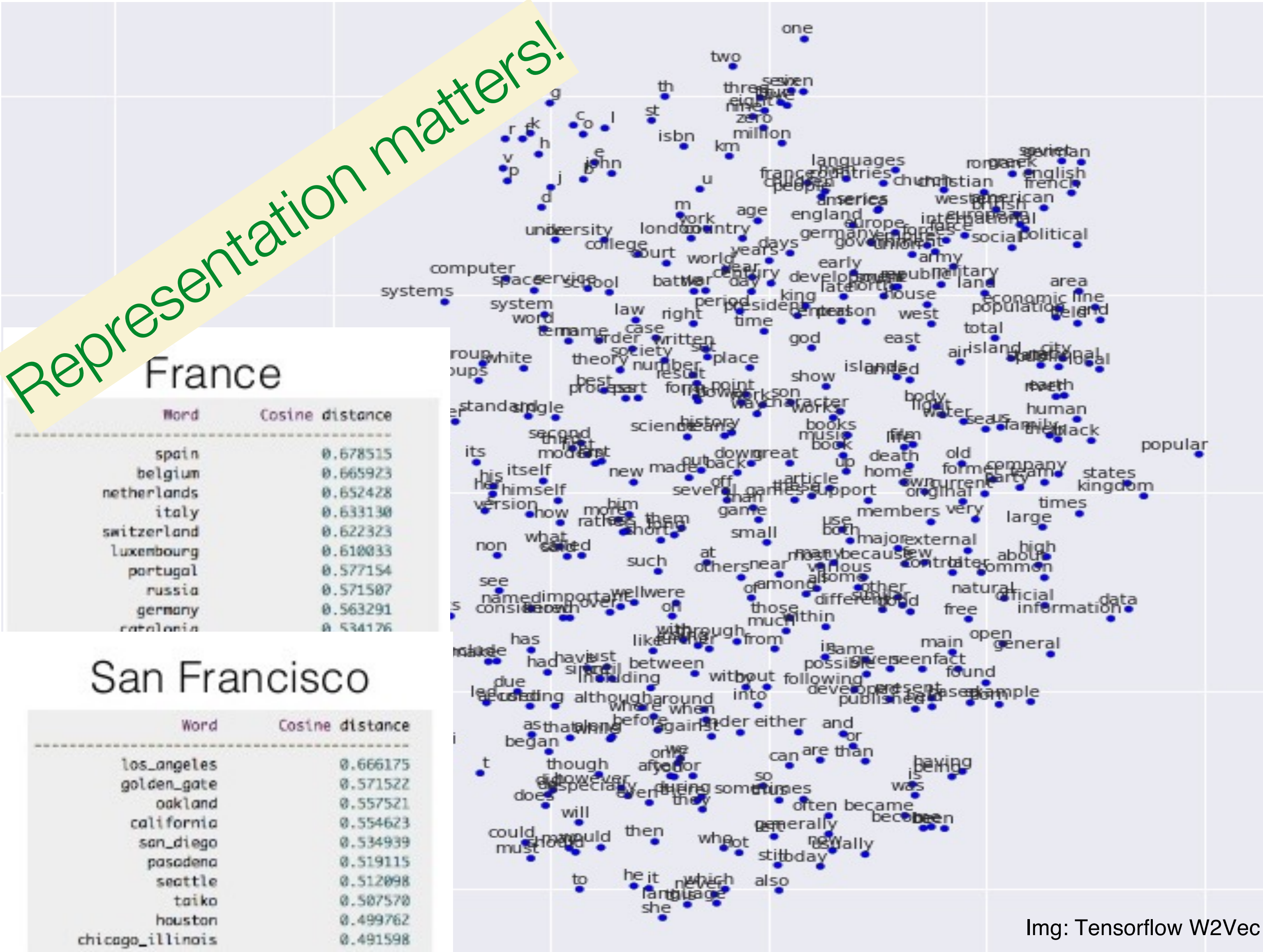
- Euclidean = L_2
- Manhattan = L_1

$$||\mathbf{x}_1 - \mathbf{x}_2||_p = \left(\sum_{i=1}^{n} |\mathbf{x}_{1,i} - \mathbf{x}_{2,i}|^p\right)^{\frac{1}{p}}$$

Representation matters!

Img: Tensorflow W2Vec

Representation matters!



Img: Tensorflow W2Vec

France

Word	Cosine distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

San Francisco

Word	Cosine distance
los_angeles	0.666175
golden_gate	0.571522
oakland	0.557521
california	0.554623
san_diego	0.534939
pasadena	0.519115
seattle	0.512098
taiko	0.507570
houston	0.499762
chicago_illinois	0.491598

Beyond KNN

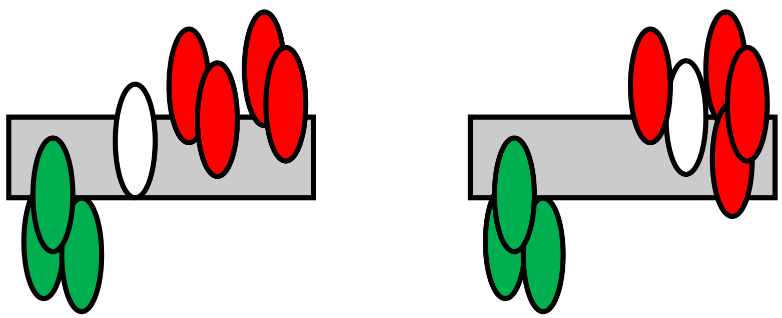
- KNN is not a statistical classifier.
- It memorizes the training data, and makes a majority vote over the K closest points.

Beyond KNN

- KNN is not a statistical classifier.
- It memorizes the training data, and makes a majority vote over the K closest points.
- For example, these two cases are the same:

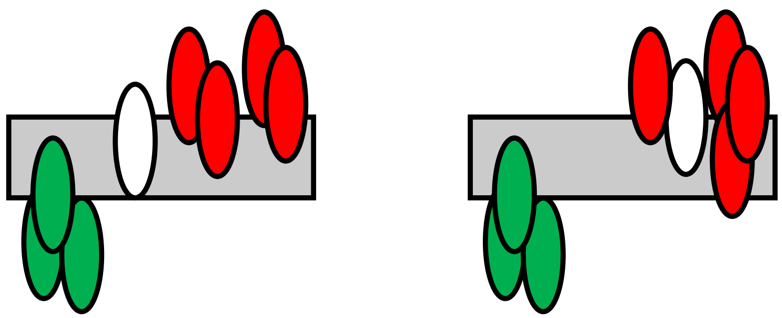
Beyond KNN

- KNN is not a statistical classifier.
- It memorizes the training data, and makes a majority vote over the K closest points.
- For example, these two cases are the same:



Beyond KNN

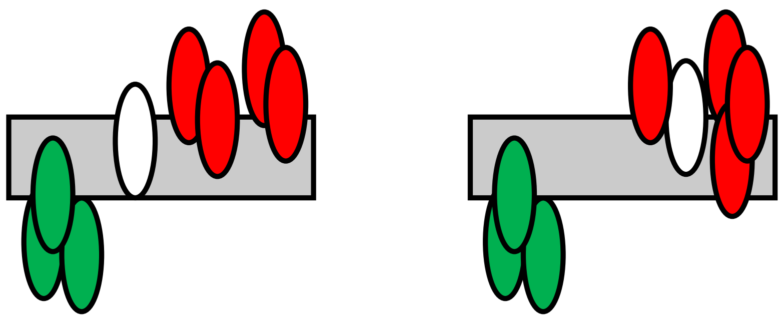
- KNN is not a statistical classifier.
- It memorizes the training data, and makes a majority vote over the K closest points.
- For example, these two cases are the same:



- What is the difference between the two scenarios?

Beyond KNN

- KNN is not a statistical classifier.
- It memorizes the training data, and makes a majority vote over the K closest points.
- For example, these two cases are the same:



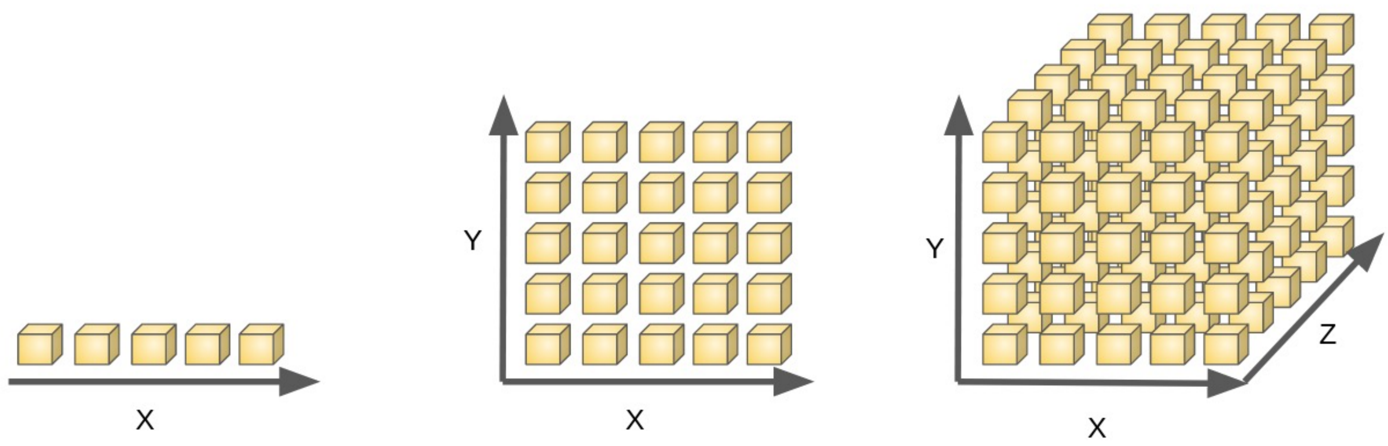
- What is the difference between the two scenarios?
- How can we reason about it?

Nearest neighbor

- Strengths:
  - Simple model, easy to implement
- Weaknesses:
  - Inefficient inference: time and space O(n)
    - (Inference time improvable with approximations, appropriate data structures)
  - Curse of dimensionality:
    - As number of features increase, you need an exponential increase in the size of the data to ensure that you have ”usable” nearby examples for any given data point

Nearest neighbor

- Strengths:
  - Simple model, easy to implement
- Weaknesses:
  - Inefficient inference: time and space O(n)
    - (Inference time improvable with approximations, appropriate data structures)
  - Curse of dimensionality:
    - As number of features increase, you need an exponential increase in the size of the data to ensure that you have ”usable” nearby examples for any given data point



kNN Can Learn ANY Function (with enough data)

- Flexibility: Nearest Neighbor rules can learn **any concept** (with enough data)

- If n training examples are sampled independently from a distribution,
- if we choose $k_n \to \infty$ as $n \to \infty$, but not too fast so $\frac{k_n}{n} \to 0$ as $n \to \infty$, then
- kNN's classifier will converge to an optimal predictor

- **This is called "universal consistency"**