Machine Learning

CSE 142

Xin (Eric) Wang

Wednesday, November 10, 2021

Today

- Distance metrics and clustering (Ch. 8)
  - K Nearest Neighbor

Notes

- Readers' OH rescheduled to Friday 1-2pm for this week
- HW2 (grades out next week)
  - **Must use your UCSC ID as the Codalab username (as described in HW2)**
  - Include your username in the HW reports
  - **Zero credits if no association** between your report and the Codalab results (email graders to fix it for HW2, but no tolerance for HW3 and HW4)
- HW3 due next Wed (November 17 midnight)
  - Finetuning your model locally
  - Only submit it to CodaLab for testing
  - You are not supposed to exploit the test set for validation, which is considered cheating in ML

Distance metrics and clustering

Chapter 8 in the textbook

Distance and clustering

- In many machine learning methods – especially geometric models – the notion of distance is important
- Especially in clustering, where we assume similarity is some function of distance
- Clustering is grouping data without prior information (unlabeled data)
- Why cluster?
  - To make apparent the natural groupings/structure in the data (perhaps for further processing)
  - To discover previously unknown relationships
  - To provide generic labels for the data

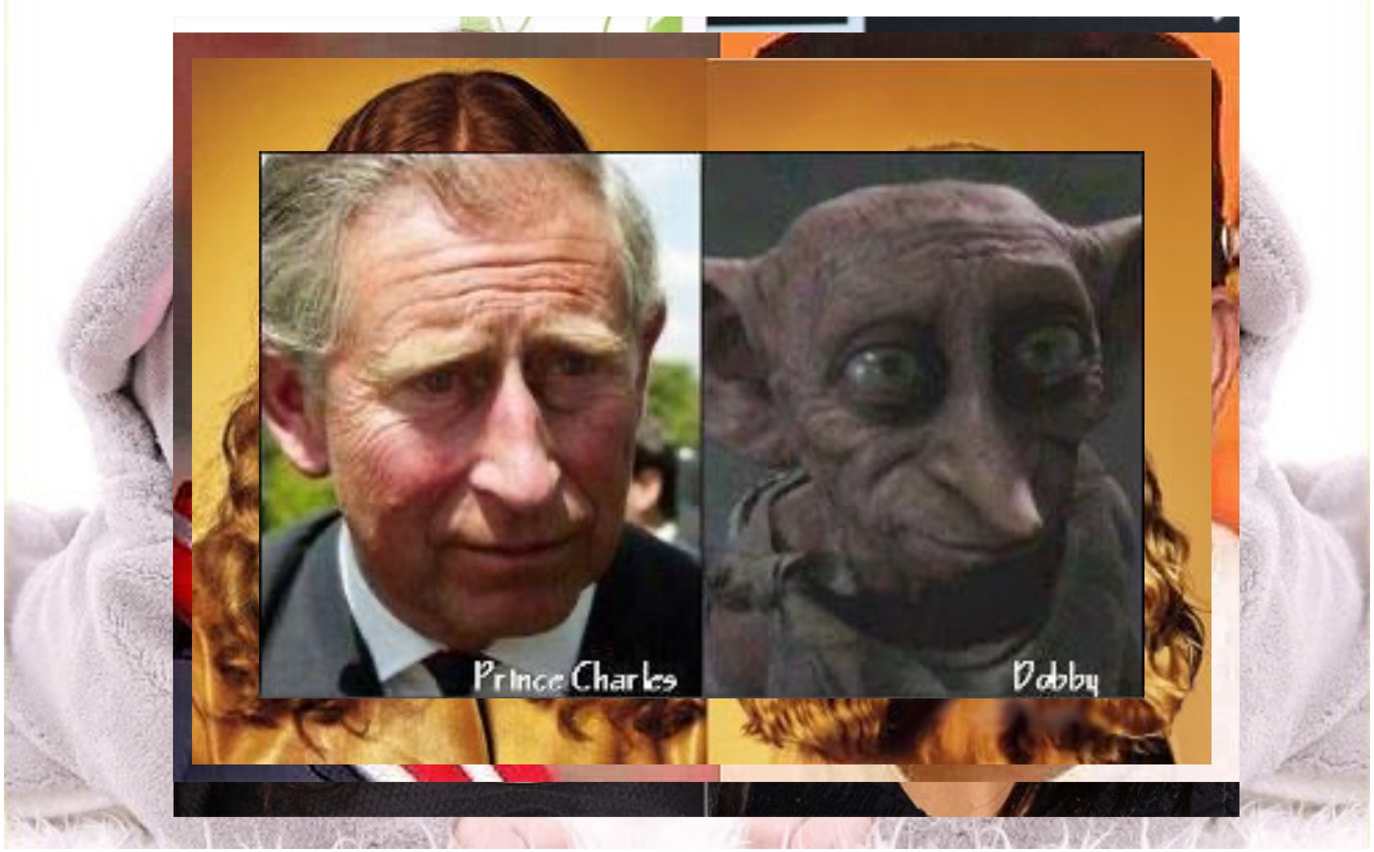
Clustering

- In clustering, we organize data into classes such that:
  - The within-class (intra-class) similarity is high
    - Lower intra-class variance
  - The between-class (inter-class) similarity is low
    - Higher inter-class variance
  - Objects in the same group (a cluster) are more similar to one another than to objects in other groups (clusters)

- But similarity and grouping may not be obvious…
- We'd like to define features and distance measures that will capture the intended notion of similarity

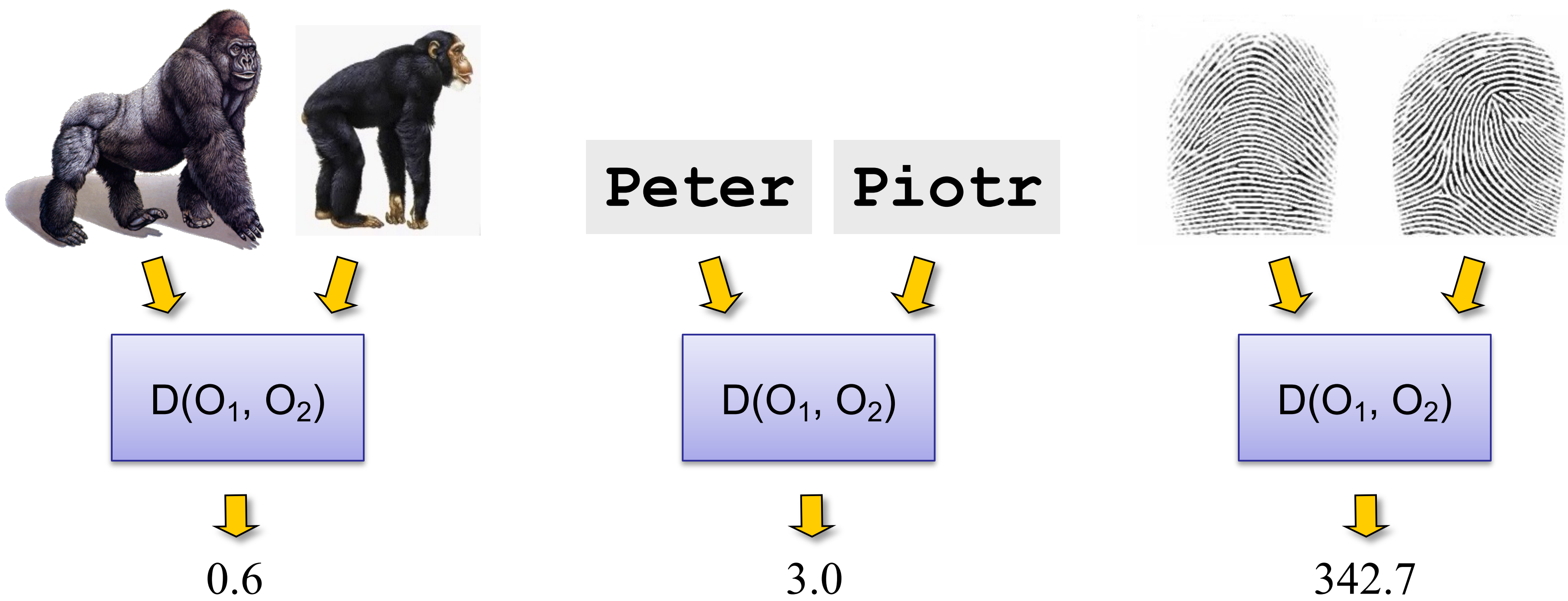
Distance $\propto$ dissimilarity

What is similarity?



Distance measures

Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



Distance measures

A distance metric $D(x_1, x_2)$ is a function $D : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that for any $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$:

1. $D(\boldsymbol{x}, \boldsymbol{x}) = 0$
2. If $\boldsymbol{x} \neq \boldsymbol{y}$ then $D(\boldsymbol{x}, \boldsymbol{y}) > 0$
3. $D(\boldsymbol{x}, \boldsymbol{y}) = D(\boldsymbol{y}, \boldsymbol{x})$ (commutative)
4. $D(\boldsymbol{x}, \boldsymbol{z}) \leq D(\boldsymbol{x}, \boldsymbol{y}) + D(\boldsymbol{y}, \boldsymbol{z})$ (triangle inequality)

Or we can refer to a norm $D(v)$ of the difference $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{y}$, such that $D : \mathcal{X} \rightarrow \mathbb{R}$:

1. $D(\mathbf{0}) = 0$
2. If $\boldsymbol{v} \neq \mathbf{0}$ then $D(\boldsymbol{v}) > 0$
3. $D(\boldsymbol{v}) = D(-\boldsymbol{v})$
4. $D(\boldsymbol{a} + \boldsymbol{b}) \leq D(\boldsymbol{a}) + D(\boldsymbol{b})$

Some common distance measures

- Manhattan (L1) distance: $D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{d} |x_i - y_i| = \|\boldsymbol{x}-\boldsymbol{y}\|_1$

  1-norm, Cityblock/Manhattan distance

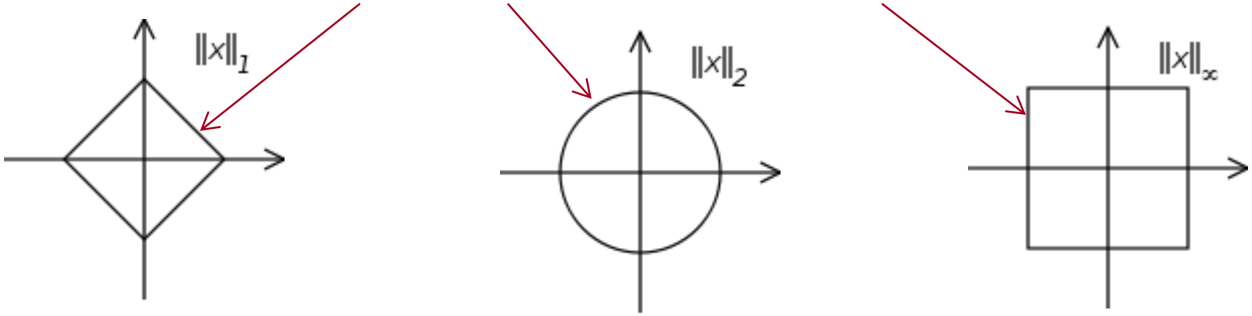
- Euclidian (L2) distance: $D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{d} (x_i - y_i)^2\right)^{1/2} = \|\boldsymbol{x}-\boldsymbol{y}\|_2$

  2-norm

- Minkowski (L_p) distance: $D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p} = \|\boldsymbol{x}-\boldsymbol{y}\|_p$

  p-norm

These points are equidistant from the origin



L1 L2 L_∞

Some common distance metrics (cont.)

- L_∞ distance/norm is known as Chebyshev distance

$$L_\infty(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_\infty = \max_i |x_i - y_i|$$

- L_0 distance/norm counts the number of non-zero elements

$$L_0(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_0 = \text{count}(|x_i - y_i| > 0)$$

  - This is the Hamming distance if $\boldsymbol{x}$ and $\boldsymbol{y}$ are binary vectors

- Mahalanobis distance takes into account the covariance in a data set

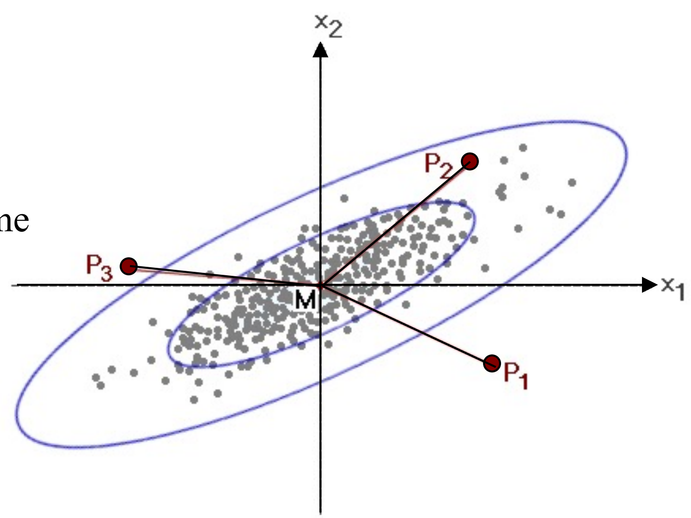
$$D_M(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{y})}$$

1. The distance between $\boldsymbol{x}$ and the origin $\boldsymbol{y}$, or
2. The distance between two variables that have the same distribution (and thus the same covariance matrix)

where $\boldsymbol{\Sigma}$ is the covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{k} \boldsymbol{X}_z \boldsymbol{X}_z{}^T = \frac{1}{k} \boldsymbol{S}$$

Scatter matrix



Distance-based methods

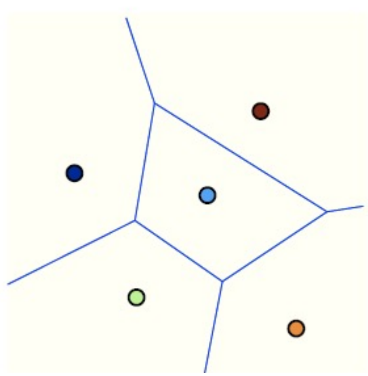
- Methods for classification and clustering based on distances to exemplars or neighbors
  - Exemplar – a prototypical instance
    - E.g., the ideal example instance of Class A
  - Neighbor – a "nearby" instance or exemplar
    - E.g., within some distance radius d

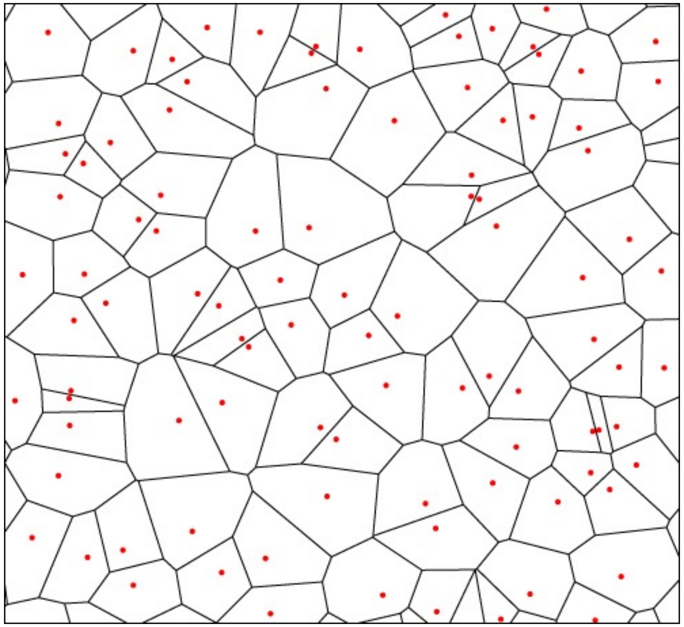
- Our basic (binary) linear classifier follows this procedure:
  1. Construct an exemplar for each class from its mean
  2. Assign a new instance $\boldsymbol{x}$ to the nearest exemplar using Euclidian distance

- This is a basic nearest neighbor (NN) approach
  - No explicit construction of a decision boundary is required

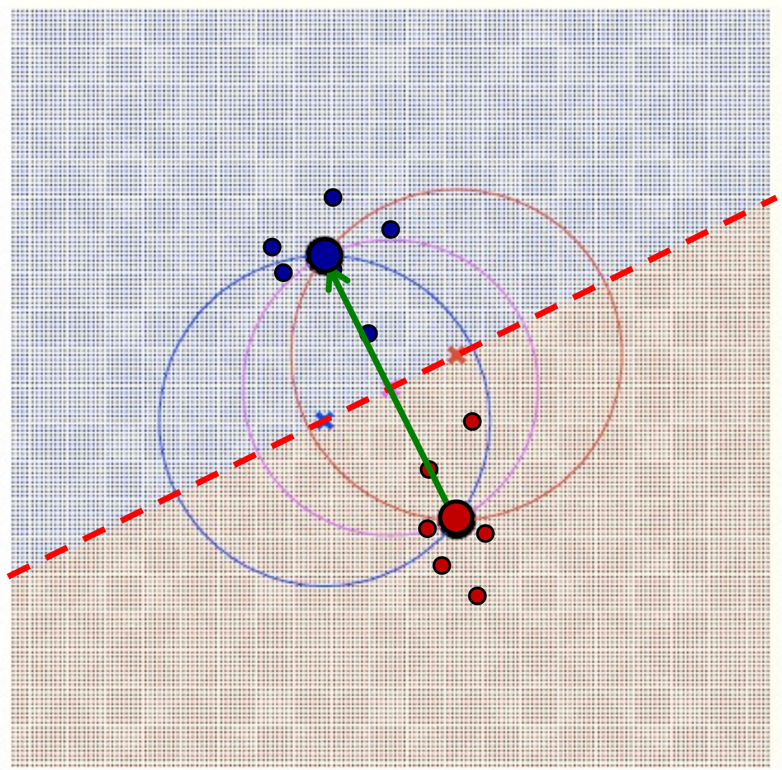
1-Nearest neighbor (1NN) classifier

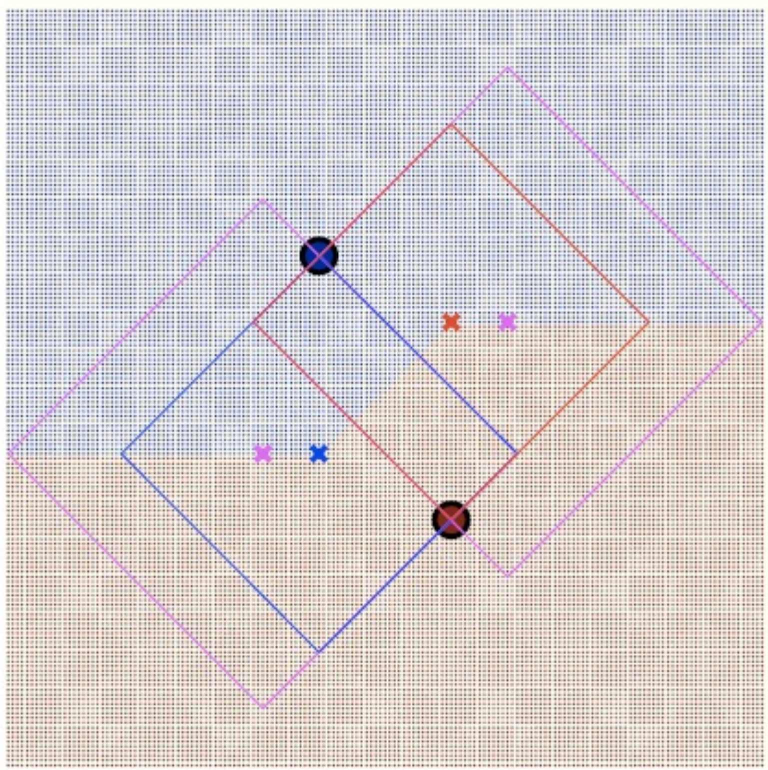
- The simplest nearest neighbor classifier: Assign the new instance ***x*** to the nearest labeled training point (or exemplar)
  - Training = memorizing the training data
  - Each point is an exemplar, or exemplars are computed from the data
  - But it generalizes, unlike the lookup table approach
  - The *implicit* decision boundaries of a 1NN classifier comprise a Voronoi diagram
    - Leads to piecewise linear decision boundaries



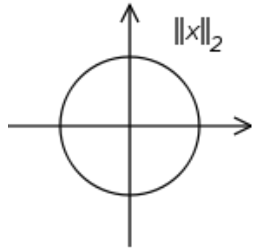


Implicit 1NN decision boundaries (N=2)

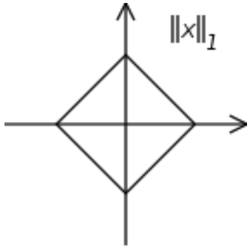




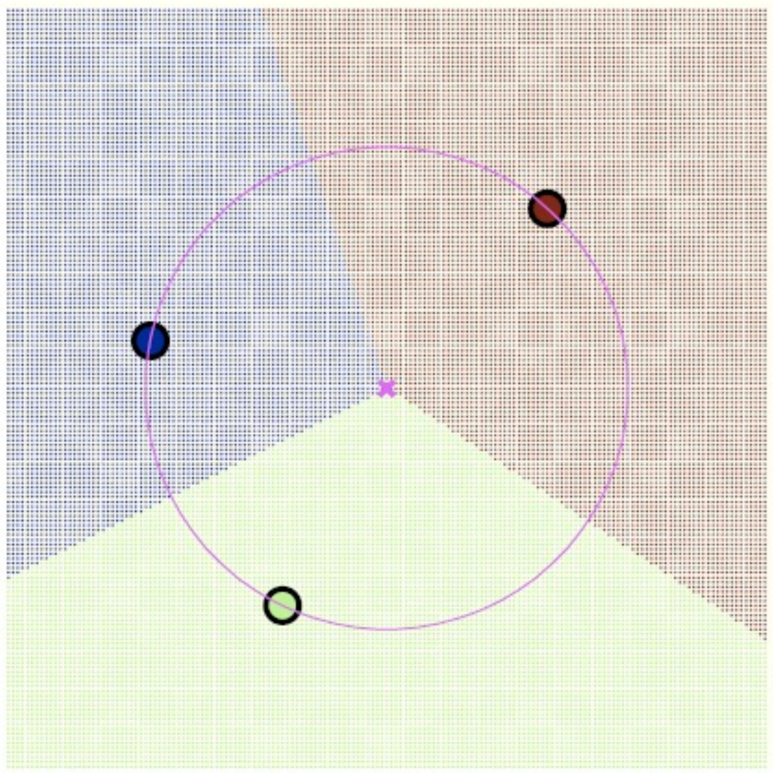
Euclidian (L2)



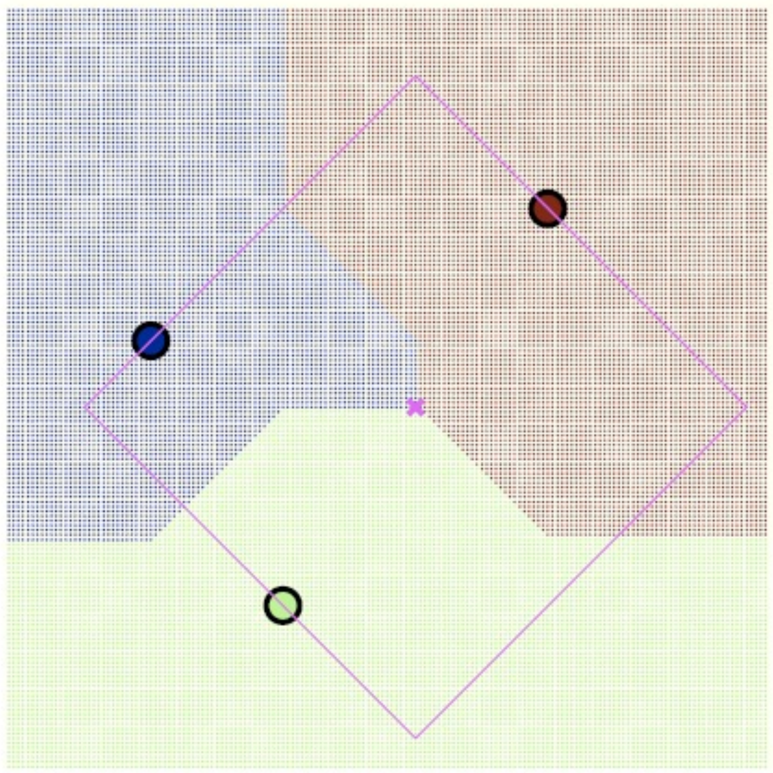
Manhattan (L1)



Implicit 1NN decision boundaries (N=3)



Euclidian (L2)



Manhattan (L1)

Multi-class version of the basic linear classifier

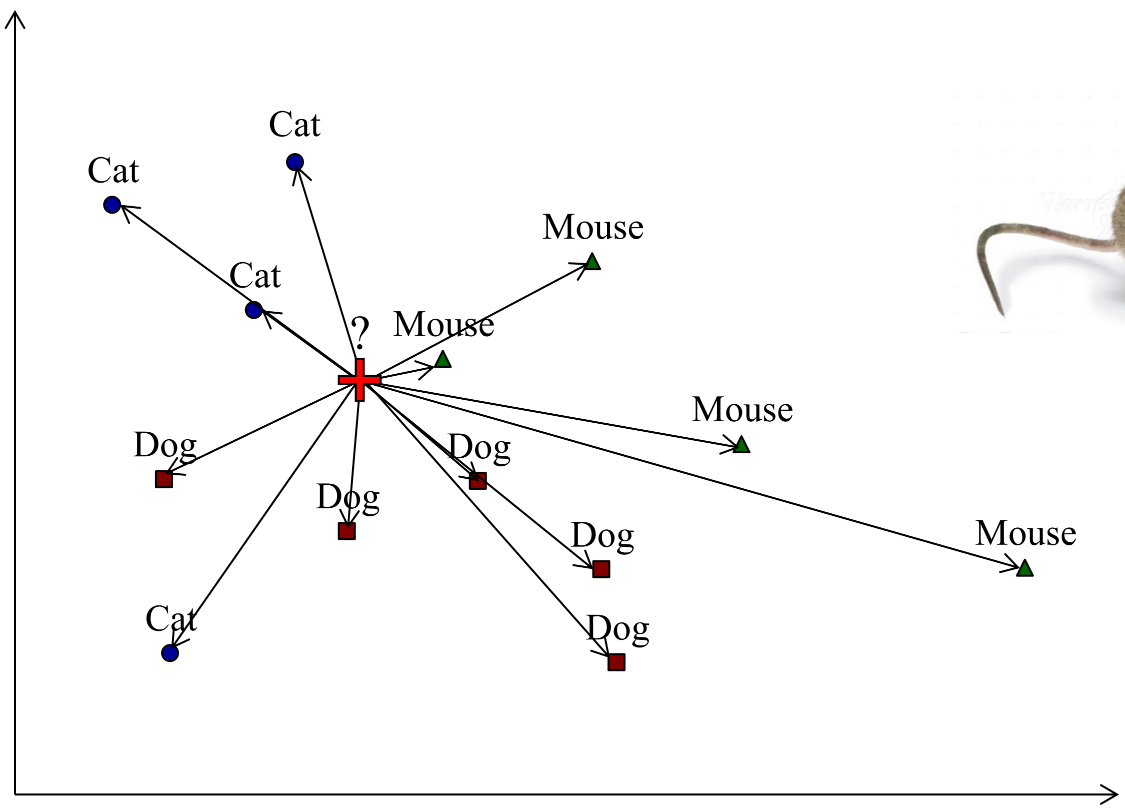
k-Nearest neighbor (*k*NN) classifiers

- In some cases, the *k-nearest neighbor* method is preferable:
  - Classify a new instance by taking a vote of the $k \geq 1$ nearest exemplars
  - E.g., in a binary classifier, with $k = 7$, for a new input point the 7 nearest neighbors may include 5 positives and 2 negatives, so we choose positive as the classification
- Or, instead of using a fixed *k*, vote among all neighbors within a fixed radius *r*
- Or, combine the two, stopping when (*count* > *k*) or (*dist.* > *r*)
- May also use distance weighting – the closer an exemplar is to the instance, the more its vote counts (e.g., $w_i = \frac{1}{D(\boldsymbol{x}, \boldsymbol{x}_i)}$)
- What about ties in the voting?
  - Preference to the 1NN
  - Random choice
  - Etc.

Nearest neighbor (1NN) classifier

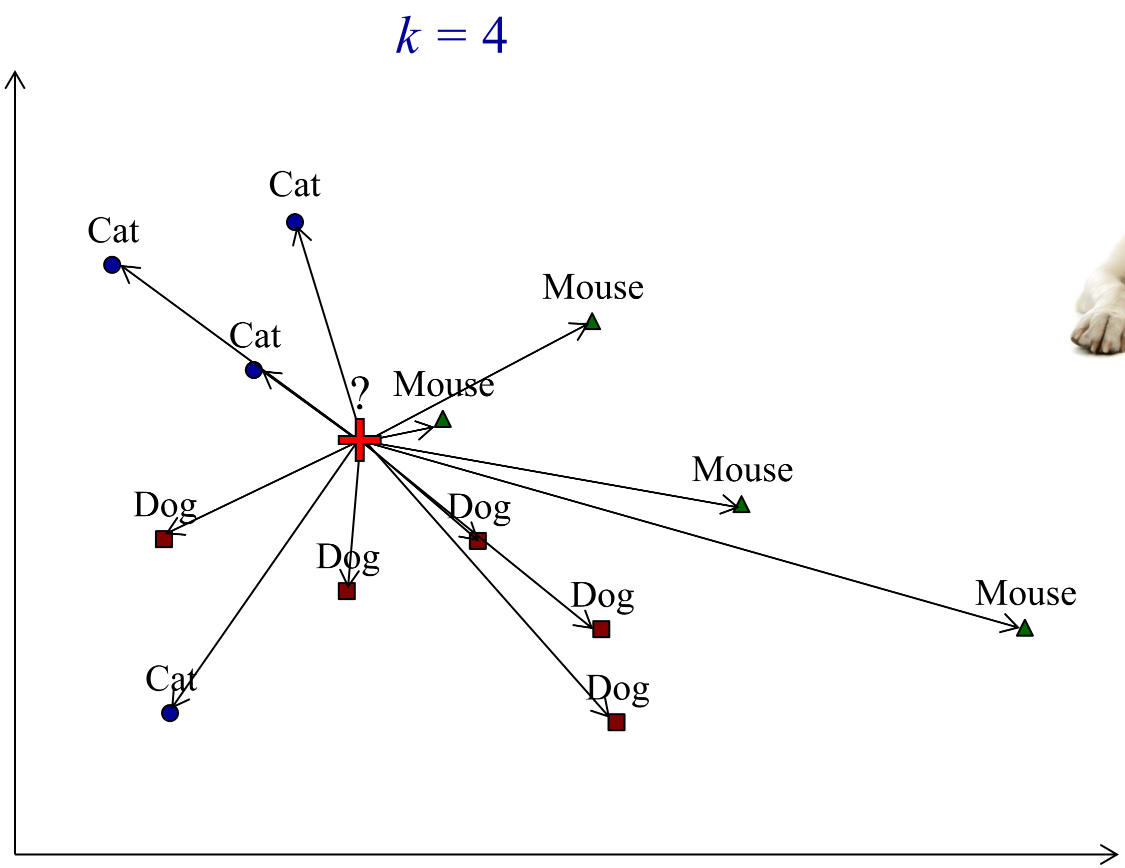
Class(x) = nearest training data point to x

- Based on some distance metric (L1, L2, etc.)



k-Nearest neighbor (kNN) classifier

Class(x) = plurality vote among k nearest training data points to x

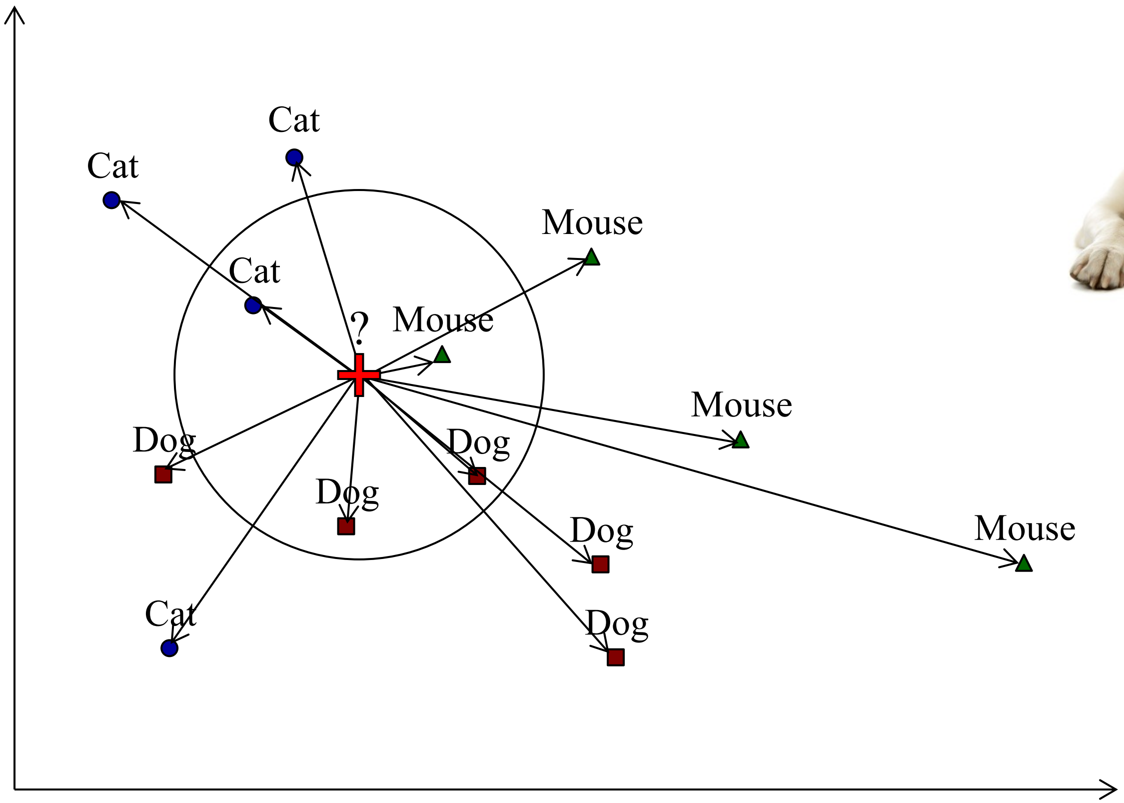


k-Nearest neighbor (kNN) classifier

Class(x) = plurality vote among nearest training data points to x within distance r

Could also weigh the voting based on distance

$r = 2.5$



Quiz: *k*-Nearest Neighbor (*k*NN)

- See the quiz on Canvas

Nearest neighbor classification – summary

- NN classifiers are very fast to train – $O(n)$ time
  - n = # of training samples
- But its classification is relatively slow – also $O(n)$ time
  - Need to compare the input instance with every stored training example (or at least every exemplar)
- Importantly, NN methods rely on a useful distance metric
  - *Nearest* in Euclidian distance, Manhattan distance, Mahalanobis distance, or what?
  - This is problem-dependent
  - Distance-based methods
- Bottom line: nearest neighbor classifiers are simple, intuitive, and train quickly
  - But they can be inefficient, may require a good deal of storage, and can't easily represent a specific boundary geometry