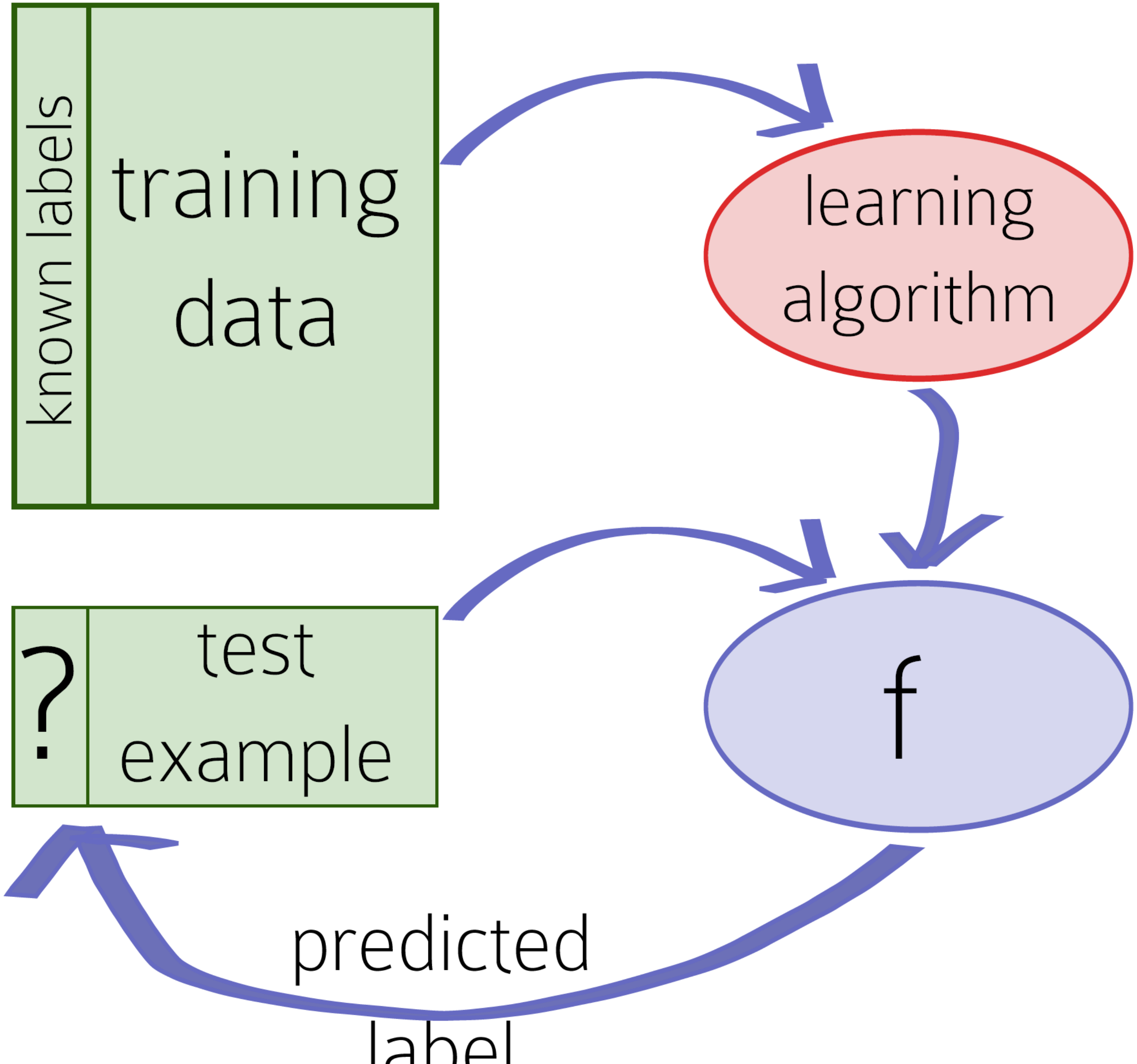
Classification with Nearest Neighbors

CMSC 422

MARINE CARPUAT
marine@cs.umd.edu


known labels
training data
learning algorithm
?
test example
f
predicted label

Law of Large Numbers

Suppose that $v_1, v_2, \ldots v_N$ are independent and identically distributed random variables

The empirical sample average approaches the population average as the number of sample goes to infinity.

$$Pr\left(\lim_{N\to\infty} \frac{1}{N} \sum_n v_n = \mathbb{E}[v]\right) = 1$$

Today's Topics

- Nearest Neighbors (NN) algorithms for classification
  - K-NN, Epsilon ball NN
- Fundamental Machine Learning Concepts
  - Decision boundary

Intuition for Nearest Neighbor Classification

This "rule of nearest neighbor" has considerable elementary intuitive appeal and probably corresponds to practice in many situations. For example, it is possible that much medical diagnosis is influenced by the doctor's recollection of the subsequent history of an earlier patient whose symptoms resemble in some way those of the current patient.

(Fix and Hodges, 1952)

Intuition for Nearest Neighbor Classification

- Simple idea
  - Store all training examples
  - Classify new examples based on most similar training examples

K Nearest Nei

K: number of neighbors that classification is based on

Test instance with unknown class in $\{-1; +1\}$

Training Data

Algorithm 3 KNN-PREDICT($\mathbf{D}$, K, $\hat{\boldsymbol{x}}$)

$S \leftarrow [\,]$
for $n = 1$ **to** N **do**
  $S \leftarrow S \oplus \langle \mathrm{d}(\boldsymbol{x}_n, \hat{\boldsymbol{x}}), n \rangle$ // store distance to training example n
end for
$S \leftarrow$ SORT(S) // put lowest-distance objects first
$\hat{y} \leftarrow 0$
for $k = 1$ **to** K **do**
  $\langle dist, n \rangle \leftarrow S_k$ // n this is the kth closest data point
  $\hat{y} \leftarrow \hat{y} + y_n$ // vote according to the label for the nth training point
end for
return SIGN($\hat{y}$) // return $+1$ if $\hat{y} > 0$ and -1 if $\hat{y} < 0$

2 approaches to learning

Eager learning

(eg decision trees)

- Learn/Train
  - Induce an **abstract model** from data
- Test/Predict/Classify
  - Apply learned model to new data

Lazy learning

(eg nearest neighbors)

- Learn
  - **Just store data** in memory
- Test/Predict/Classify
  - Compare new data to stored data
- Properties
  - Retains all information seen in training
  - Complex hypothesis space
  - Classification can be very slow

Components of a k-NN Classifier

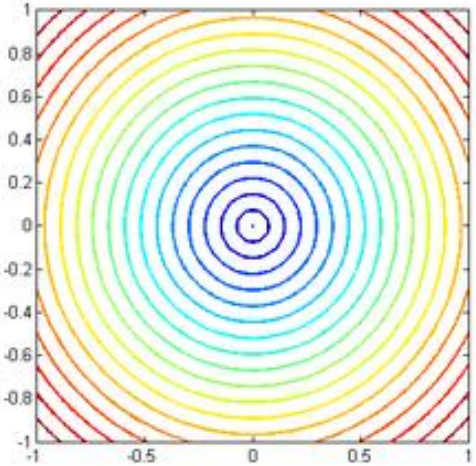
- Distance metric
  - How do we measure distance between instances?
  - Determines the layout of the example space

- The k hyperparameter
  - How large a neighborhood should we consider?
  - Determines the complexity of the hypothesis space

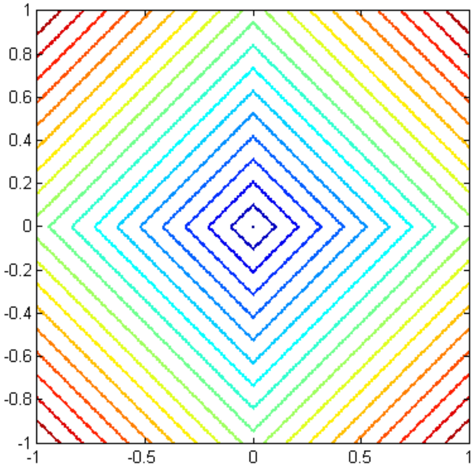
Distance metrics

- We can use any distance function to select nearest neighbors.
- Different distances yield different neighborhoods

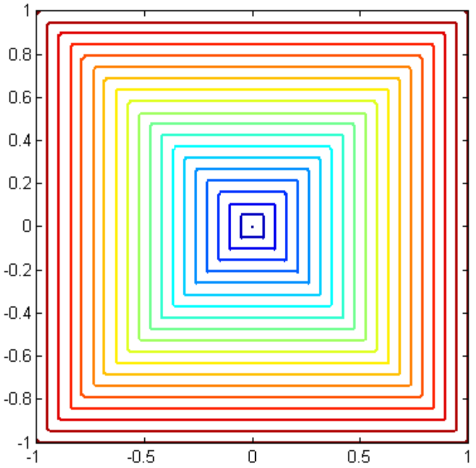
L2 distance
(= Euclidean distance)



L1 distance



Max norm



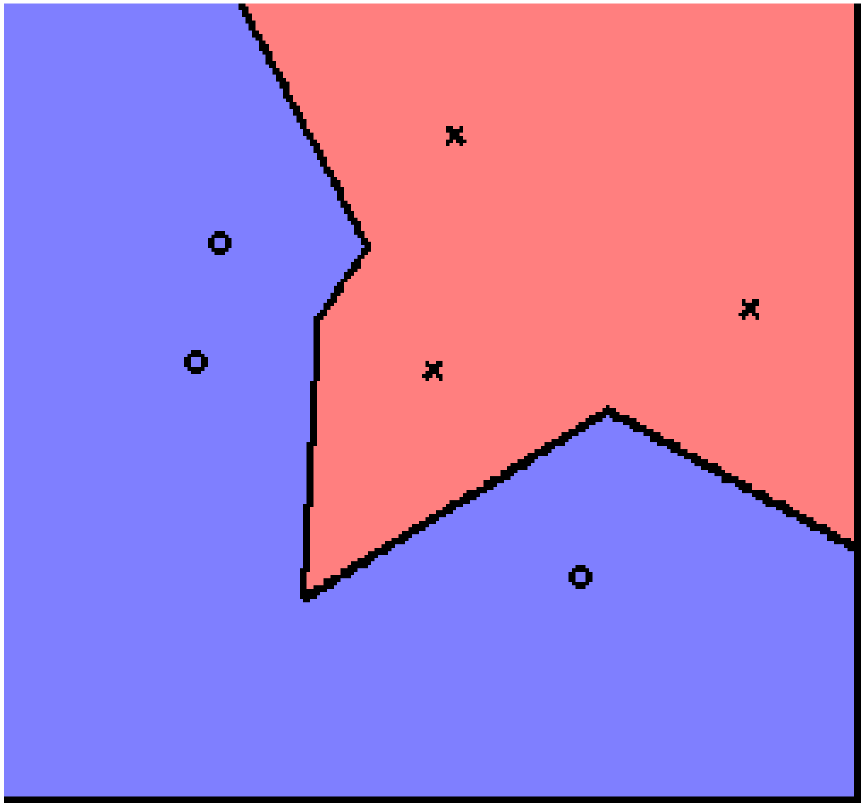
Decision Boundary of a Classifier

- is the line that separates positive and negative regions in the feature space

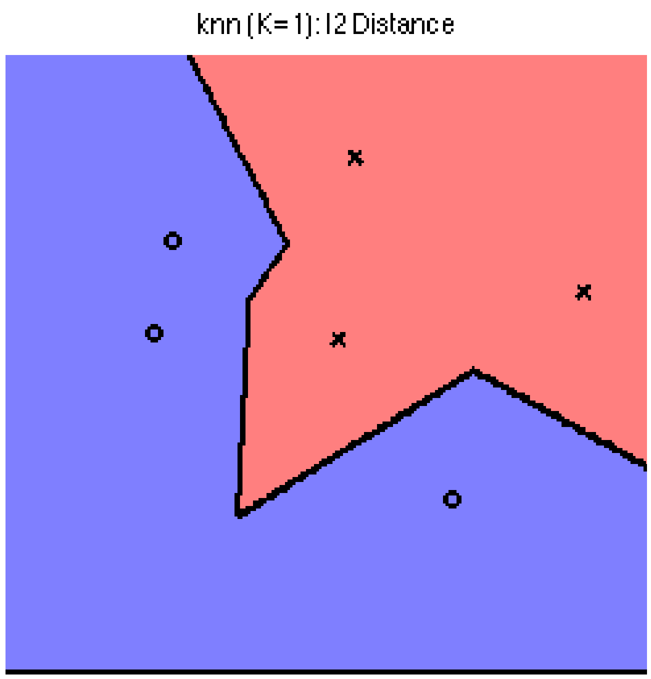
- Why is it useful?
  - it helps us visualize how examples will be classified for the entire feature space
  - it helps us visualize the complexity of the learned model

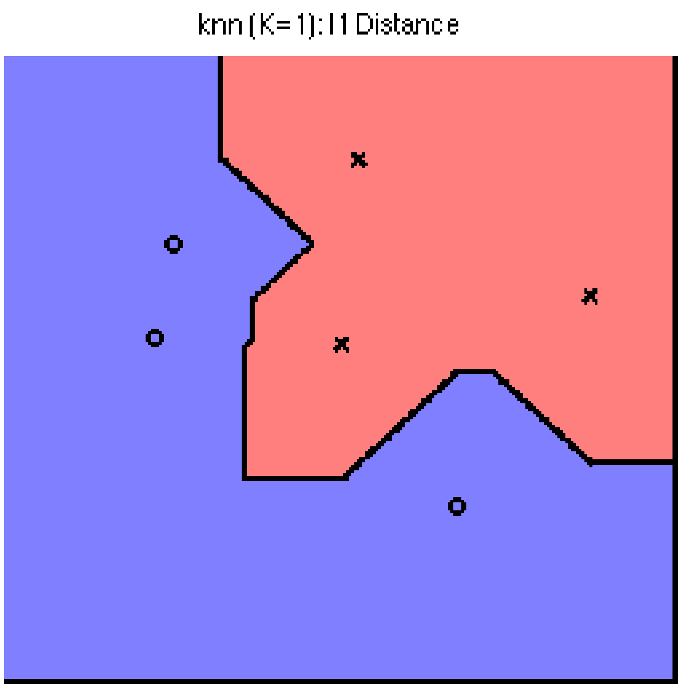
Decision Boundaries for 1-NN

knn (K=1): l2 Distance



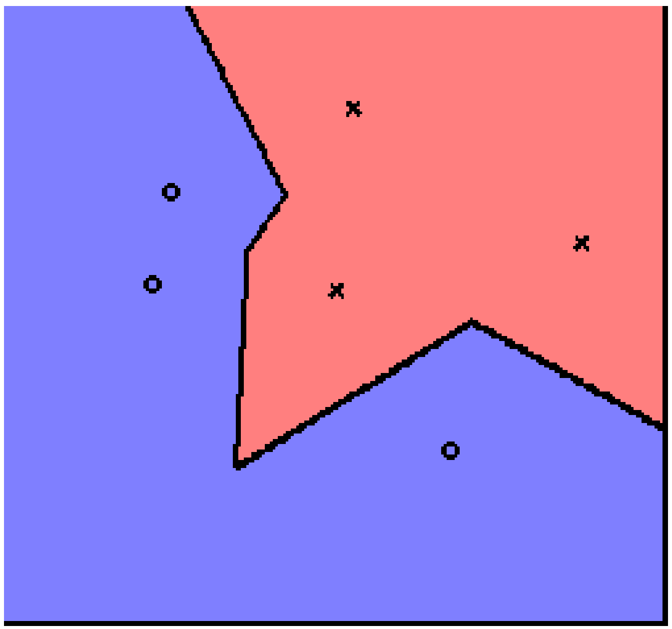
Decision Boundaries change with the distance function



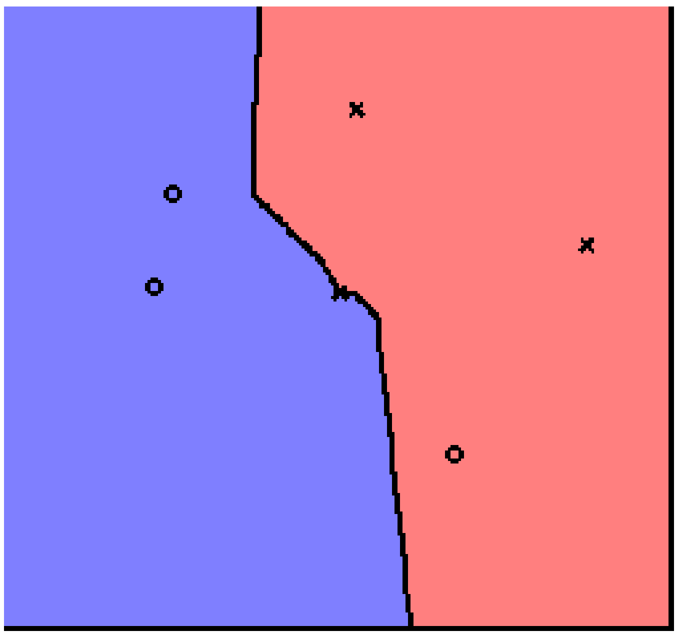


Decision Boundaries change with K

knn (K=1): l2 Distance



knn (K=3): l2 Distance



The k hyperparameter

- Tunes the complexity of the hypothesis space
  - If k = 1, every training example has its own neighborhood
  - If k = N, the entire feature space is one neighborhood!
- Higher k yields smoother decision boundaries
- How would you set k in practice?

What is the inductive bias of k-NN?

- Nearby instances should have the same label
- All features are equally important
- Complexity is tuned by the k parameter

Variations on k-NN: Weighted voting

- Weighted voting
  - Default: all neighbors have equal weight
  - Extension: weight neighbors by (inverse) distance

Variations on k-NN: Epsilon Ball Nearest Neighbors

- Same general principle as K-NN, but change the method for selecting which training examples vote

- Instead of using K nearest neighbors, use all examples x such that

$$distance(\hat{x}, x) \leq \varepsilon$$

Exercise: How would you modify KNN-Predict to perform Epsilon Ball NN?

Algorithm 3 KNN-PREDICT($\mathbf{D}$, K, $\hat{\boldsymbol{x}}$)

$S \leftarrow [\,]$
for $n = 1$ **to** N **do**
  $S \leftarrow S \oplus \langle \mathrm{d}(\boldsymbol{x}_n, \hat{\boldsymbol{x}}), n \rangle$ // store distance to training example n
end for
$S \leftarrow \textsc{sort}(S)$ // put lowest-distance objects first
$\hat{y} \leftarrow 0$
for $k = 1$ **to** K **do**
  $\langle dist, n \rangle \leftarrow S_k$ // n this is the kth closest data point
  $\hat{y} \leftarrow \hat{y} + y_n$ // vote according to the label for the nth training point
end for
return $\textsc{sign}(\hat{y})$ // return $+1$ if $\hat{y} > 0$ and -1 if $\hat{y} < 0$

Exercise: When are DT vs kNN appropriate?

Properties of classification problem	**Can Decision Trees handle them?**	**Can K-NN handle them?**
Binary features		
Numeric features		
Categorical features		
Robust to noisy training examples		
Fast classification is crucial		
Many irrelevant features		
Relevant features have very different scale		