k-Nearest Neighbors

CSC 461: Machine Learning

Fall 2021

Prof. Marco Alvarez
University of Rhode Island

Instance-based learning

- Class of learning methods
  - ✓ also called **lazy learning**
- No need to learn any explicit hypothesis
- **Training** is trivial (just store instances)
- **Predicting** new labels is where computation happens

what is the computational complexity of training?

Nearest Neighbor Classification

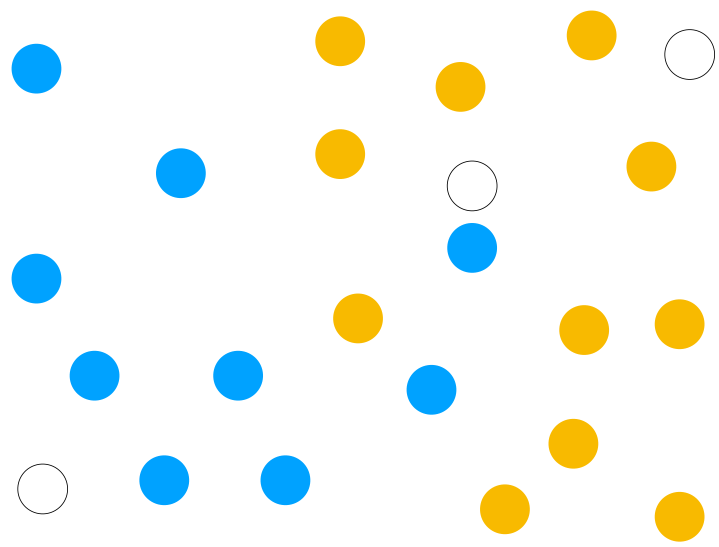
Nearest neighbor classification

- Training examples are vectors with a class label

$$x_i \in \mathbb{R}^d \qquad y_i \in \{1, \ldots, C\}$$

- Learning
  - ✓ **store** all training examples
- Prediction
  - ✓ predict the label of the new example as the label of its **closest point** in the training set

what is the computational complexity of predicting a new label?



k-Nearest Neighbors

k-nearest neighbors

- Prediction for a test point x
  - ✓ recover a subset Sx (k nearest neighbors to x)

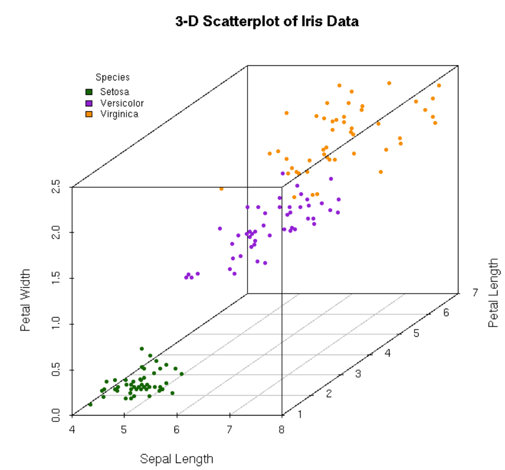
$$S_x \subseteq \mathscr{D} \text{ s.t. } |S_x| = k$$

$$\forall (\mathbf{x}', y') \in \mathscr{D}\backslash S_x$$

$$D(\mathbf{x}, \mathbf{x}') \geq \max_{(\mathbf{x}'', y'') \in S_x} D(\mathbf{x}, \mathbf{x}'')$$

  - ✓ take a **majority vote (mode)** (classification)
  - ✓ calculate the **average** (regression)

Classification example



https://spin.atomicobject.com/2013/05/06/k-nearest-neighbor-racket/

Distance

$$D(a,b) = \left(\sum_{i=1}^{d} |a_i - b_i|^p \right)^{1/p}$$

minkowski

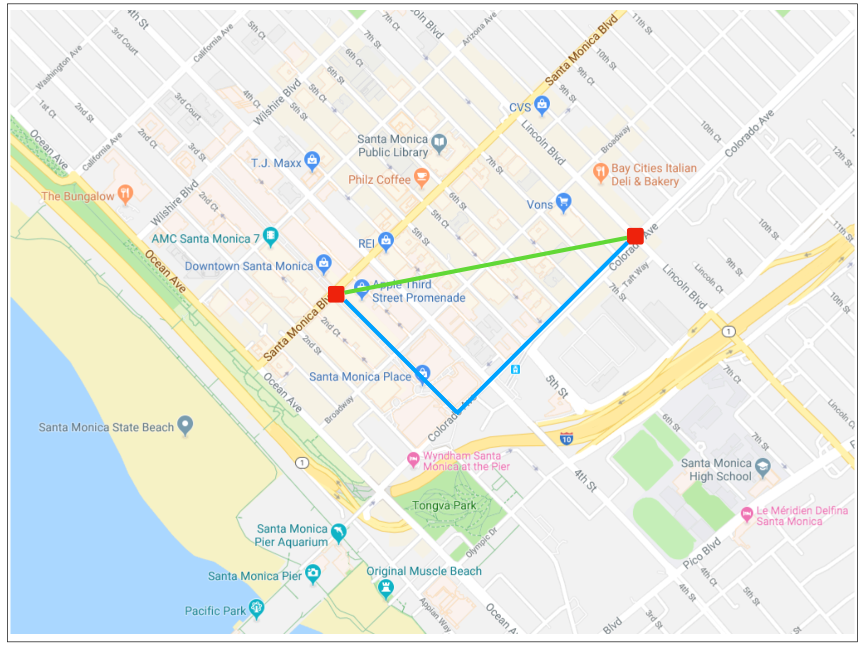
$$a \in \mathbb{R}^d, b \in \mathbb{R}^d$$

$p = 1$? manhattan

$p = 2$? euclidean

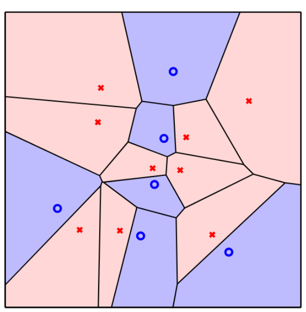
$p = \infty$? chebyshev

could also use other distances (for different input spaces)



What is the decision boundary?

- Is k-NN building an explicit decision boundary?
  - ✓ not really, but it can be inferred

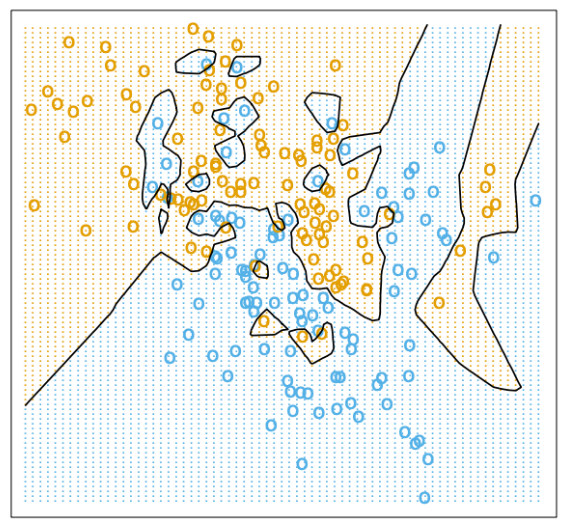


Nearest neighbor Voronoi tesselation

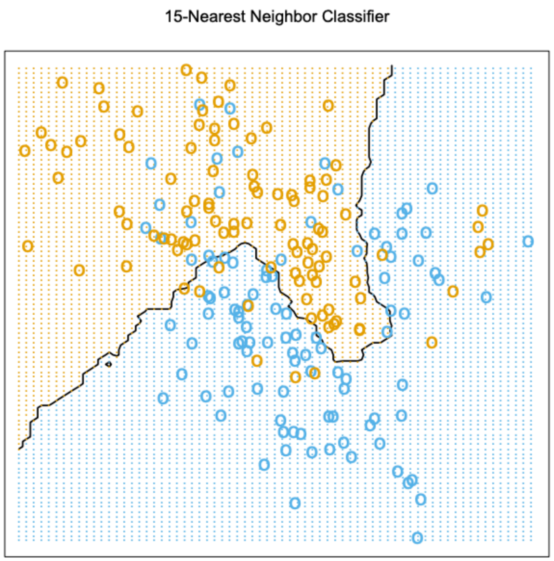
is the diagram sensitive to k? what about the distance function?

http://www.cs.rpi.edu/~magdon/courses/LFD-Slides/SlidesLect16.pdf

1-Nearest Neighbor Classifier

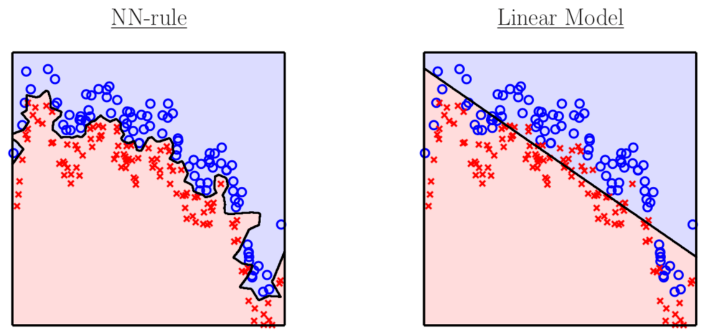


Elements of Statistical Learning (2nd Ed.) c Hastie, Tibshirani & Friedman 2009 Chap 2



Elements of Statistical Learning (2nd Ed.) c Hastie, Tibshirani & Friedman 2009 Chap 2

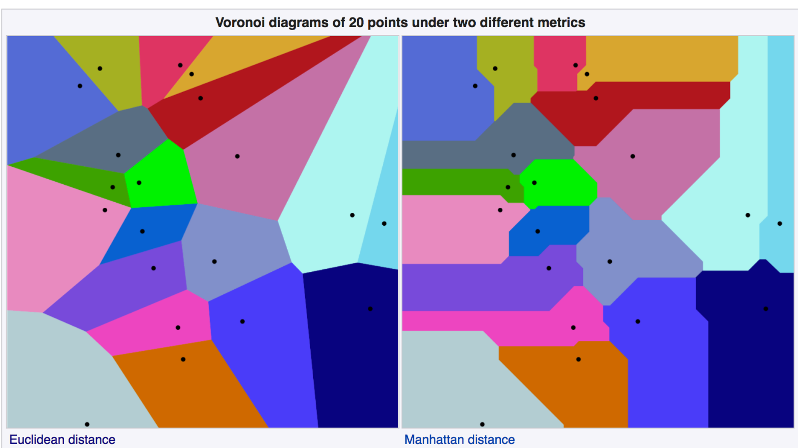
kNN vs linear models



NN-rule	Linear Model
no parameters	$(d+1)$ parameters
expressive/flexible	rigid, always linear
$g(\mathbf{x})$ needs data	$g(\mathbf{x})$ needs only weights
generic, can model anything	specialized

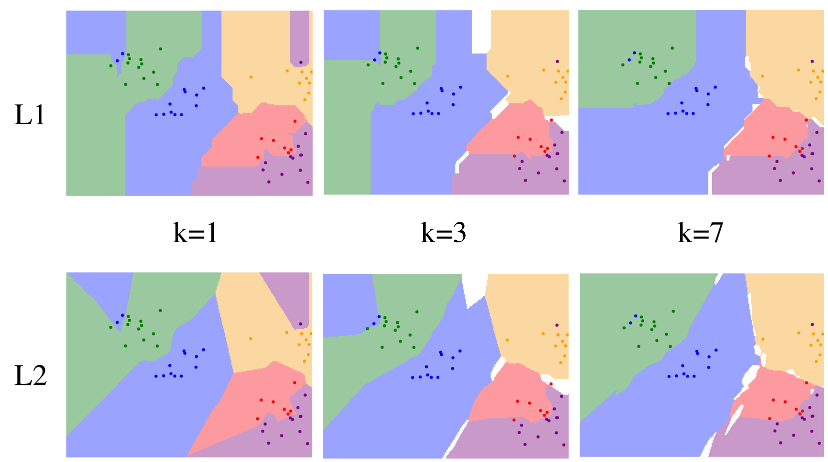
http://www.cs.rpi.edu/~magdon/courses/LFD-Slides/SlidesLect16.pdf

Euclidean vs Manhattan



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

Hyperparameters



http://vision.stanford.edu/teaching/cs231n-demos/knn

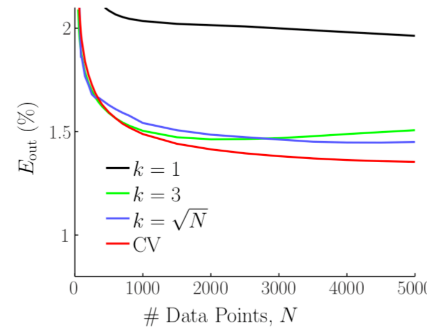
Hyperparameters

- The number of neighbors **k**
  - ✓ too small, sensitive to noise
  - ✓ too large, neighborhood includes points from other classes
- **Distance** function
- How to find a value that may generalize better?

  use Cross-Validation for parameter tuning

Choosing k

1. $k = 3$.
2. $k = \left\lceil \sqrt{N} \right\rceil$.
3. Validation or cross validation:
   k-NN rule hypotheses g_k^- constructed on training set, tested on validation set, and best k is picked.



http://www.cs.rpi.edu/~magdon/courses/LFD-Slides/SlidesLect16.pdf

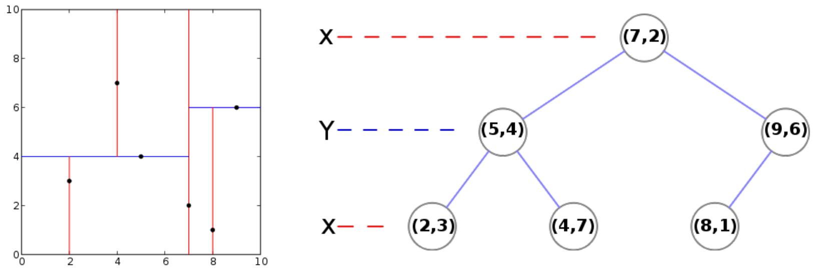
Additional Remarks

Weighted k-NN

- Can weight the votes according to distance
  - ✓ for example:

$$w = \frac{1}{d^2}$$

More efficient search



k-d Trees

Final comments

- No assumptions about **P**
  - ✓ adapts to data density
- Cost of learning is zero
  - ✓ unless a **kd-tree** or other data structures are used
- Need to normalize/scale the data
  - ✓ features with larger ranges dominate distances (automatically becoming more important)
  - ✓ be careful: sometimes range matters

Final comments

- Irrelevant or correlated attributes add noise to distance
  - ✓ may want to drop them
- Prediction is computationally expensive
  - ✓ can use **kd-trees** or **hashing techniques** like Locality Sensitive Hashing (LSH)
- Curse of dimensionality
  - ✓ data required to generalize grows exponentially with dimensionality
  - ✓ distances less meaningful in higher dimensions