kNN

10 Sep 2022

Vineeth N Balasubramanian

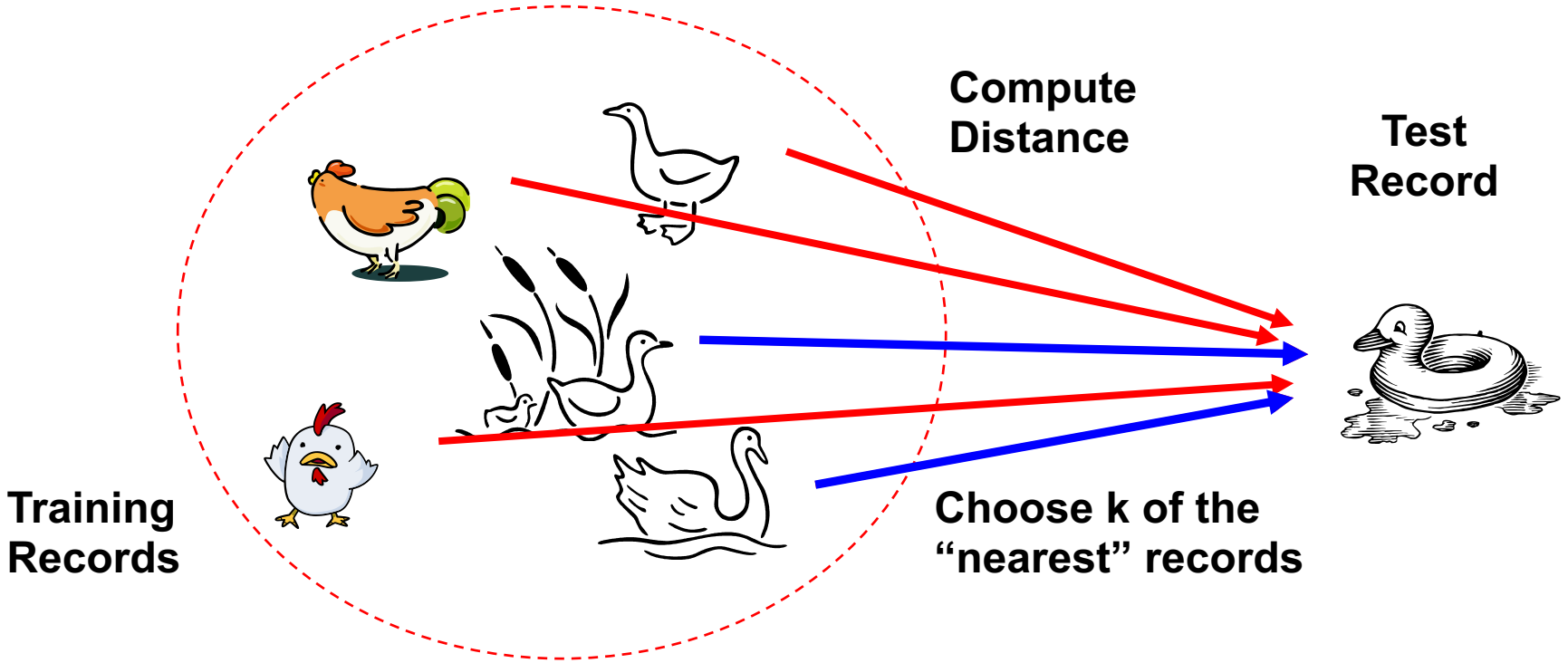
आई आई टी हैदराबाद
IIT Hyderabad

Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

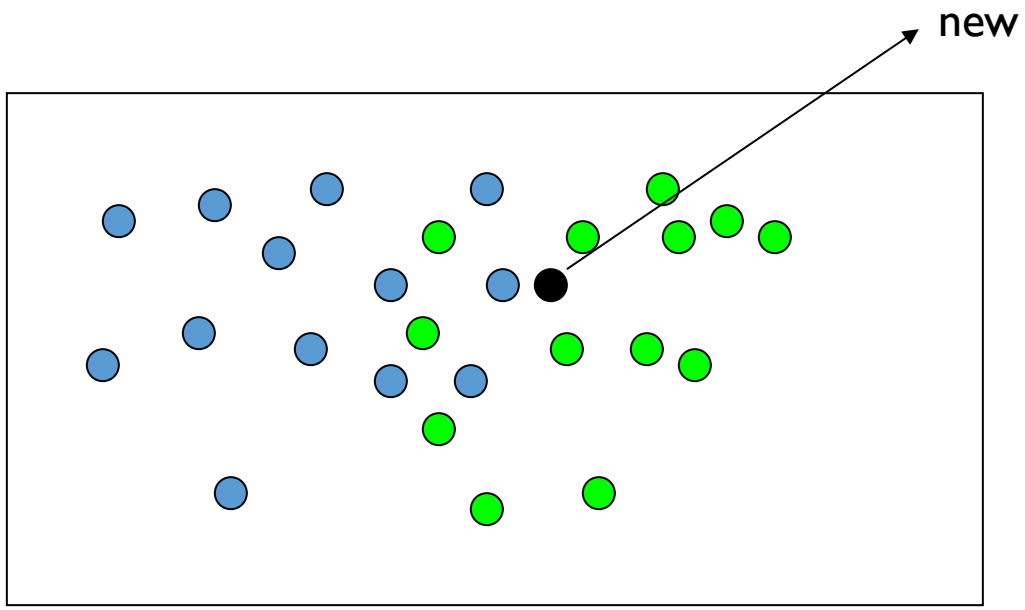
k-Nearest Neighbors

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



k-Nearest Neighbors

- Majority vote within the k nearest neighbors



K= 1: blue
K= 3: green

k-Nearest Neighbors

- An arbitrary instance is represented by $(a_1(x), a_2(x), a_3(x),.., a_n(x))$
  - $a_i(x)$ denotes features
- Euclidean distance between two instances
  - $d(x_i, x_j)=sqrt\ (sum\ for\ r=1\ to\ n\ (a_r(x_i) - a_r(x_j))^2)$
- L_p distance
  - p=2: Euclidean distance
  - p=1: Manhattan distance
  - p =∞: Max distance
  - p= 0: Count non-zero distance
- In case of continuous-valued target function
  - Mean value of k nearest training examples

Other Distance Metrics

- Cosine Distance Metric

$$\rho(\vec{x}_1, \vec{x}_2) = \cos(\angle(\vec{x}_1, \vec{x}_2)) = \frac{\vec{x}_1 \cdot \vec{x}_2}{\|\vec{x}_1\|_2 \ \|\vec{x}_2\|_2}$$

- Edit Distance

x_1 = AAATCCCGTAA

x_2 = AATCGCGTAA

Minimum number of insertions, deletions and mutations needed

$$\rho(x_1, x_2) = 2$$

- Kendell-Tau Distance

x_1 = [o1 o2 o3 o4 o5]

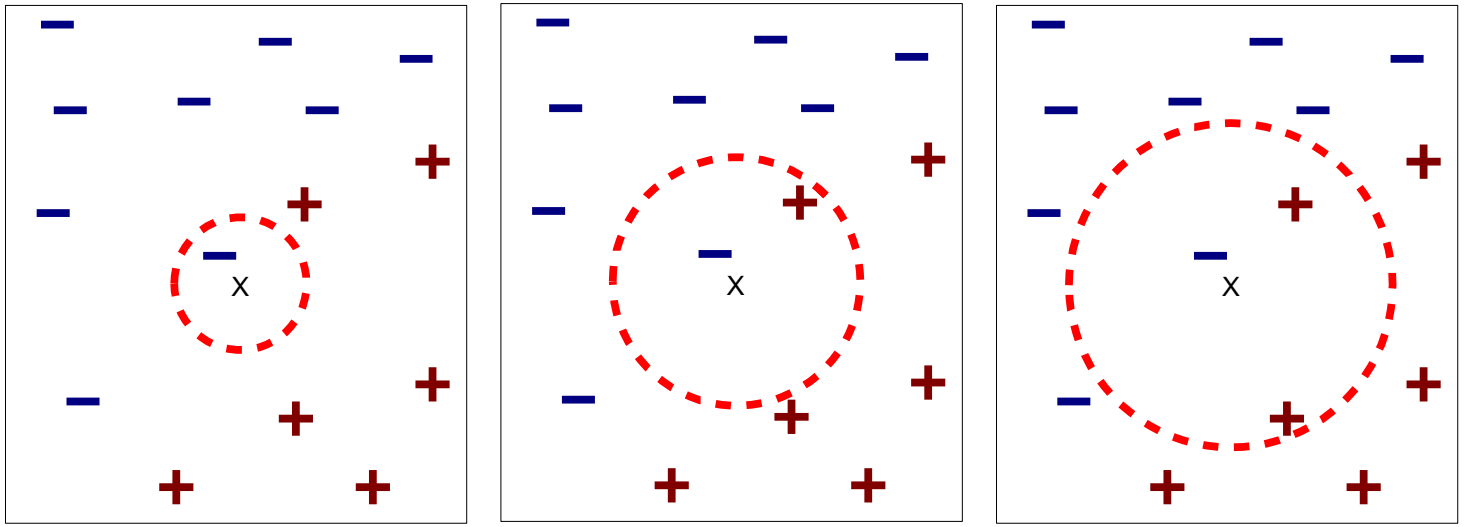
x_2 = [o2 o1 o3 o4 o5]

Bubble sort distance to make one ranking order same as the other

$$\rho(x_1, x_2) = 1$$

k-Nearest Neighbors

- Choosing k is important
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

How to determine k

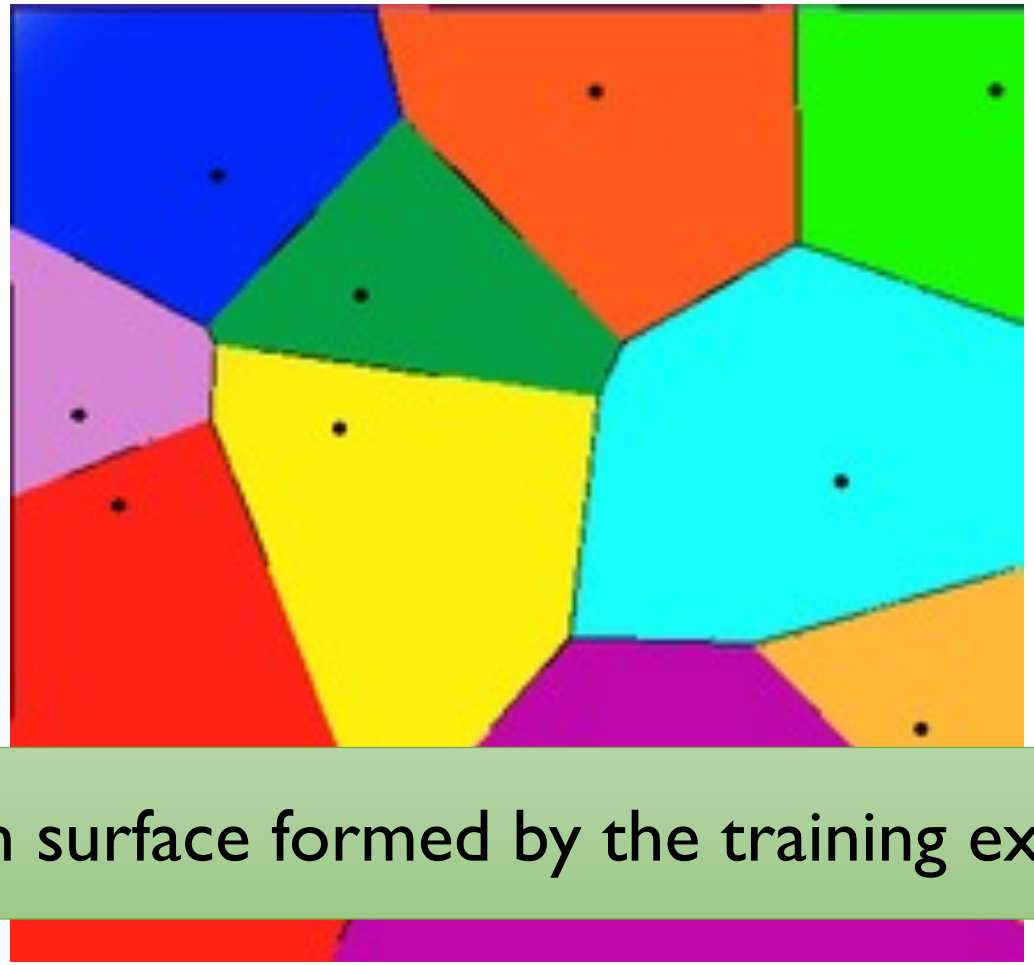
- Determined experimentally (think cross-validation!)
  - Start with k=1 and use a test set to validate the error rate of the classifier
  - Repeat with k=k+2
  - Choose the value of k for which the error rate is minimum
  - Note: k typically an odd number to avoid ties in binary classification

k-Nearest Neighbors

- Eager Learning (Induction)
  - Explicit description of target function on the whole training set
- Instance-based Learning (Transduction)
  - Learning=storing all training instances
  - Classification=assigning target function to a new instance
  - Referred to as "Lazy" learning

Similar Keywords: K-Nearest Neighbors, Memory-Based Reasoning, Example-Based Reasoning, Instance-Based Learning, Case-Based Reasoning, Lazy Learning

Voronoi Diagram



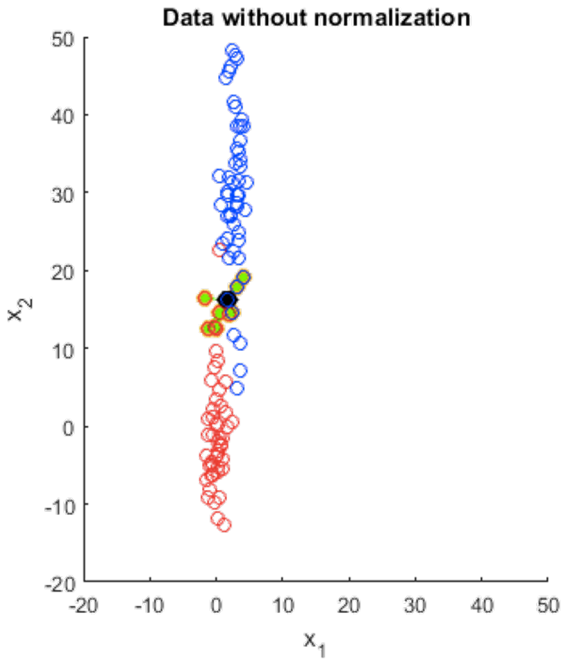
Decision surface formed by the training examples!

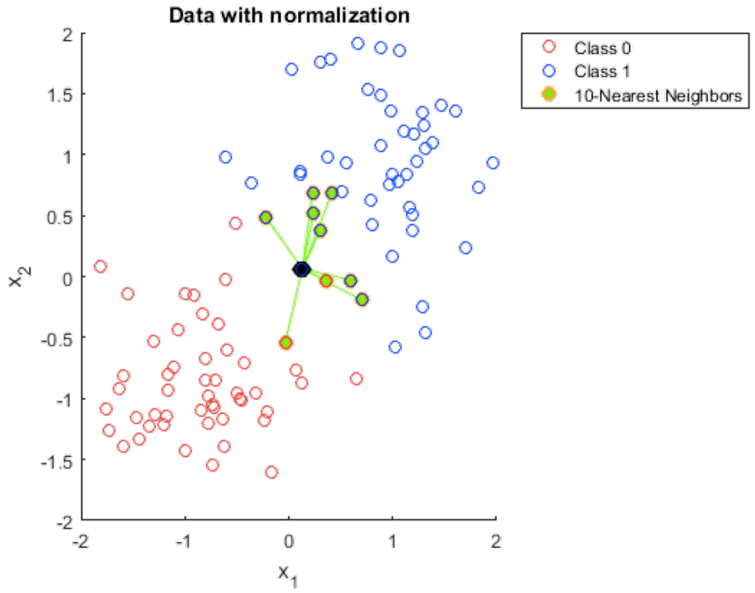
Pros and Cons

- Pros
  - Highly effective transductive inference method for noisy training data and complex target functions
  - Target function for a whole space may be described as a combination of less complex local approximations
  - Trains very fast ("Lazy" learner)
- Cons
  - Curse of dimensionality
    - In higher dimensions, all data points lie on the surface of the unit hypersphere!
  - Closeness in raw measurement space may not be good for the task
  - Storage: all training examples are saved in memory
    - A decision tree or linear classifier is much smaller
  - Slow at query time
    - Can be overcome and presorting and indexing training samples

Improvements

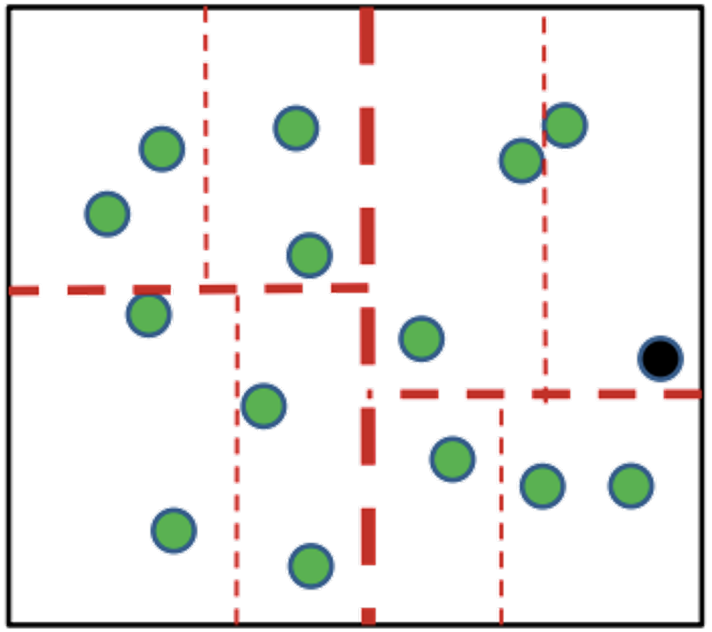
- Measure "closeness" differently
- Distance-Weighted Nearest Neighbors
  - Assign weights to the neighbors based on their 'distance' from the query point (E.g., weight 'may' be inverse square of the distances)
  - Can also learn this -> "**Metric Learning**"
- Scaling (**normalization**) attributes for fair computation of distances





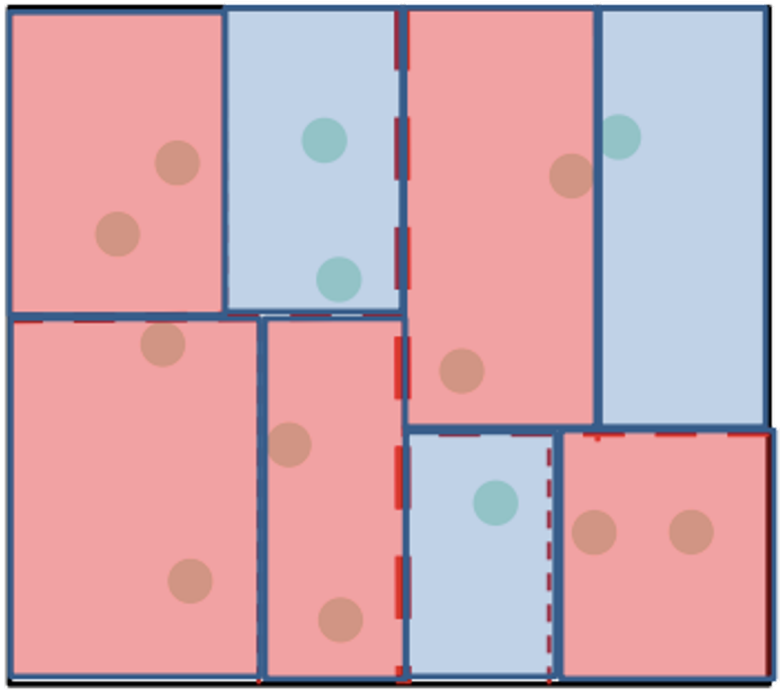
Improvements

- Finding "close" examples in a large training set quickly
  - E.g. Efficient memory indexing using kd-trees
    - In 1-dimension, can reduce complexity from O(n) to O(log n) – assuming data is sorted
  - Other methods
    - Cover trees, Ball trees, spill trees
    - Locality-Sensitive Hashing, Vector Quantization, Clustering-based methods



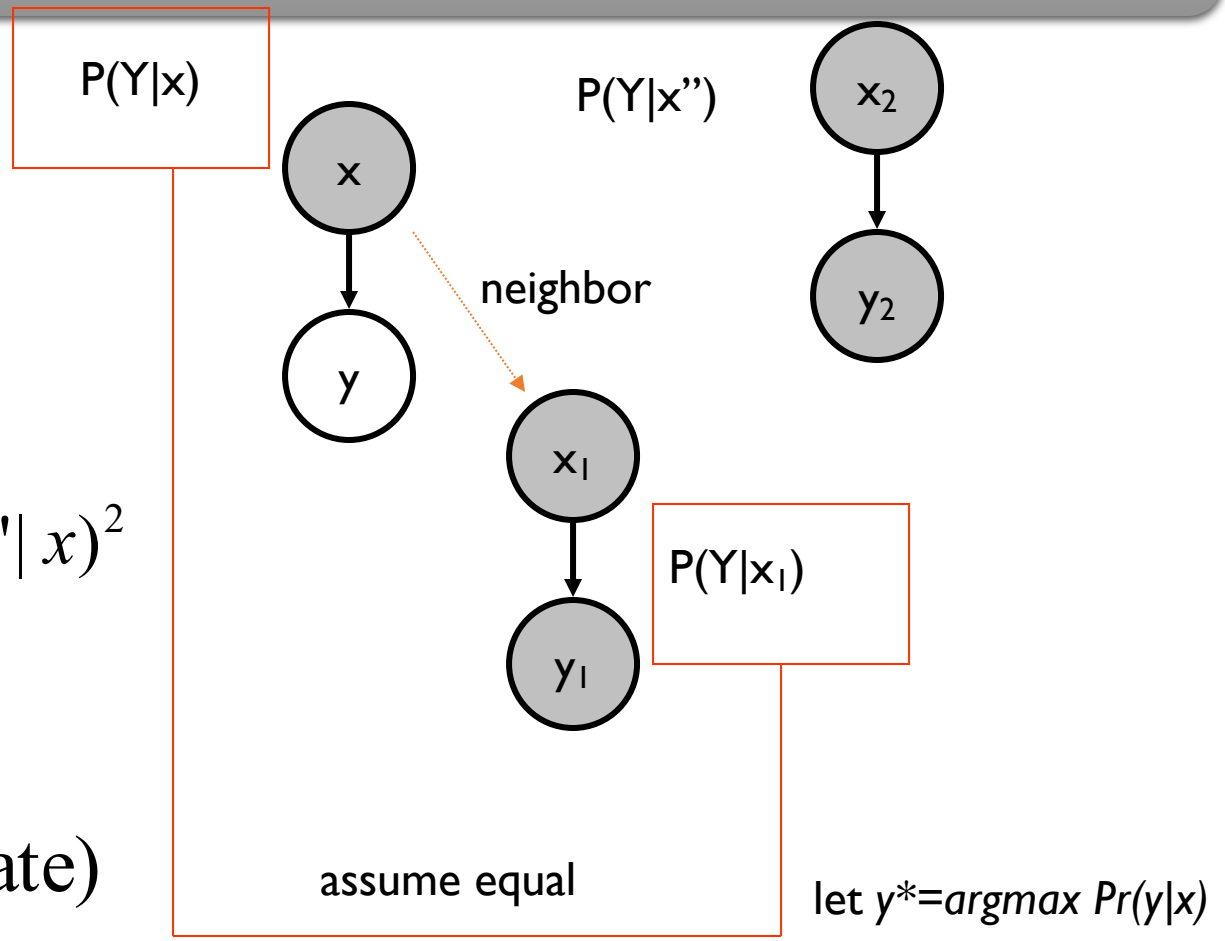
Improvements

- Not storing all examples
  - We can label each cell instead and discard the training data
  - What's the space requirement then?
  - # cells (of width r) = $\min\{n, \sim(1/r)^d\}$



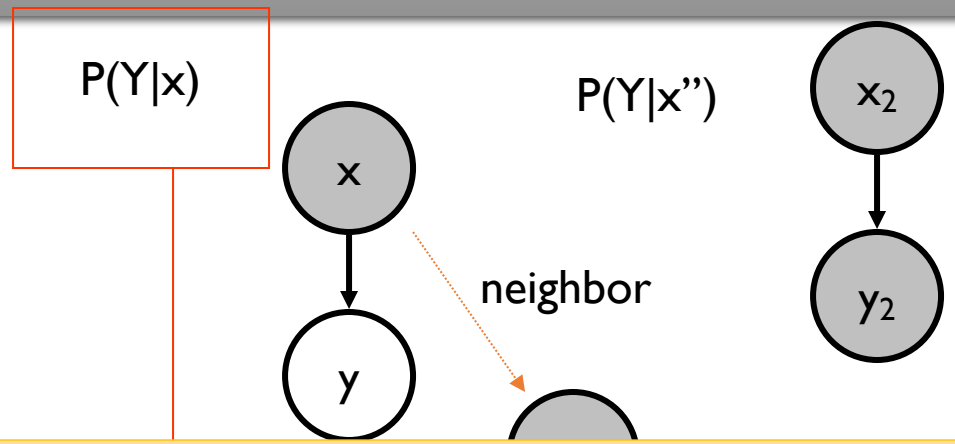
Convergence of 1-NN

$$
\begin{aligned}
&P(\text{knnError}) \\
&= 1 - \Pr(y = y_1) \\
&= 1 - \sum_{y'} \Pr(Y = y' \mid x)^2 \\
&= 1 - \Pr(y^* \mid x)^2 - \sum_{y' \neq y^*} \Pr(Y = y' \mid x)^2 \\
&\ldots \\
&\leq 2(1 - \Pr(y^* \mid x)) \\
&= 2(\text{Bayes optimal error rate})
\end{aligned}
$$



Convergence of 1-NN

$$P(\text{knnError})$$
$$= 1 - \Pr(y = y_1)$$
$$= 1 - \sum_{y'} \Pr(Y = y' | x)^2$$



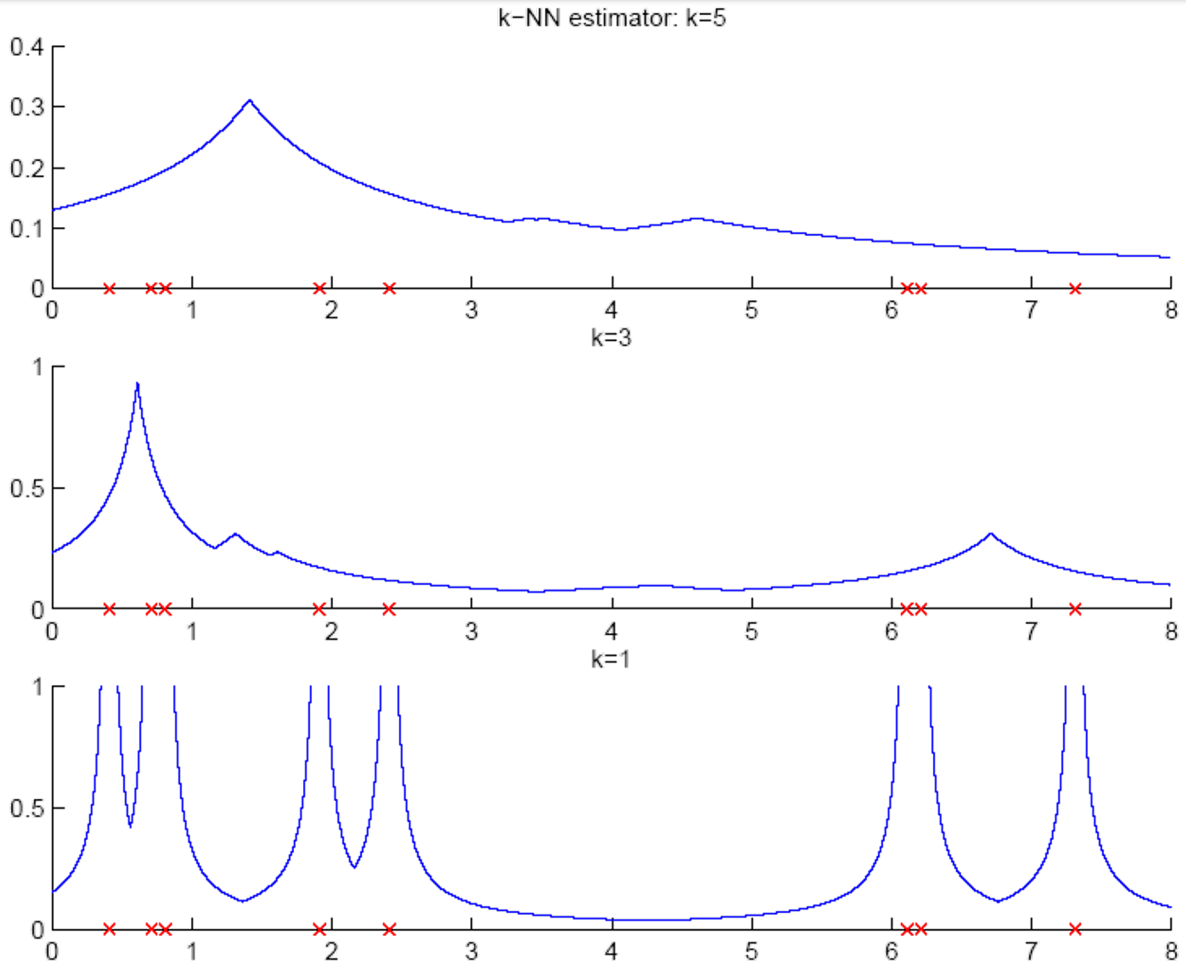
Possible to show that: as the size of training data set approaches infinity, the one nearest neighbor classifier guarantees an error rate of no worse than twice the Bayes error rate (the minimum achievable error rate given the distribution of the data).

Non-parametric Density Estimation using kNNs

- **K-Nearest Neighbor estimator**
- Instead of fixing bin width h and counting the number of instances, fix the instances (neighbors) k and check bin width

$$\hat{p}(x) = \frac{k}{2Nd_k(x)}$$

$d_k(x)$, distance to kth closest instance to x



Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Readings

- Chapters 8, "Introduction to Machine Learning" by Ethem Alpaydin, 2nd edition
- Chapter 2 (Sec 2. 5), Bishop, PRML