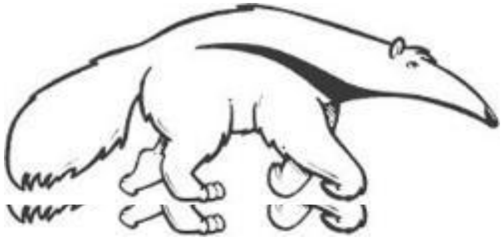
Machine Learning and Data Mining

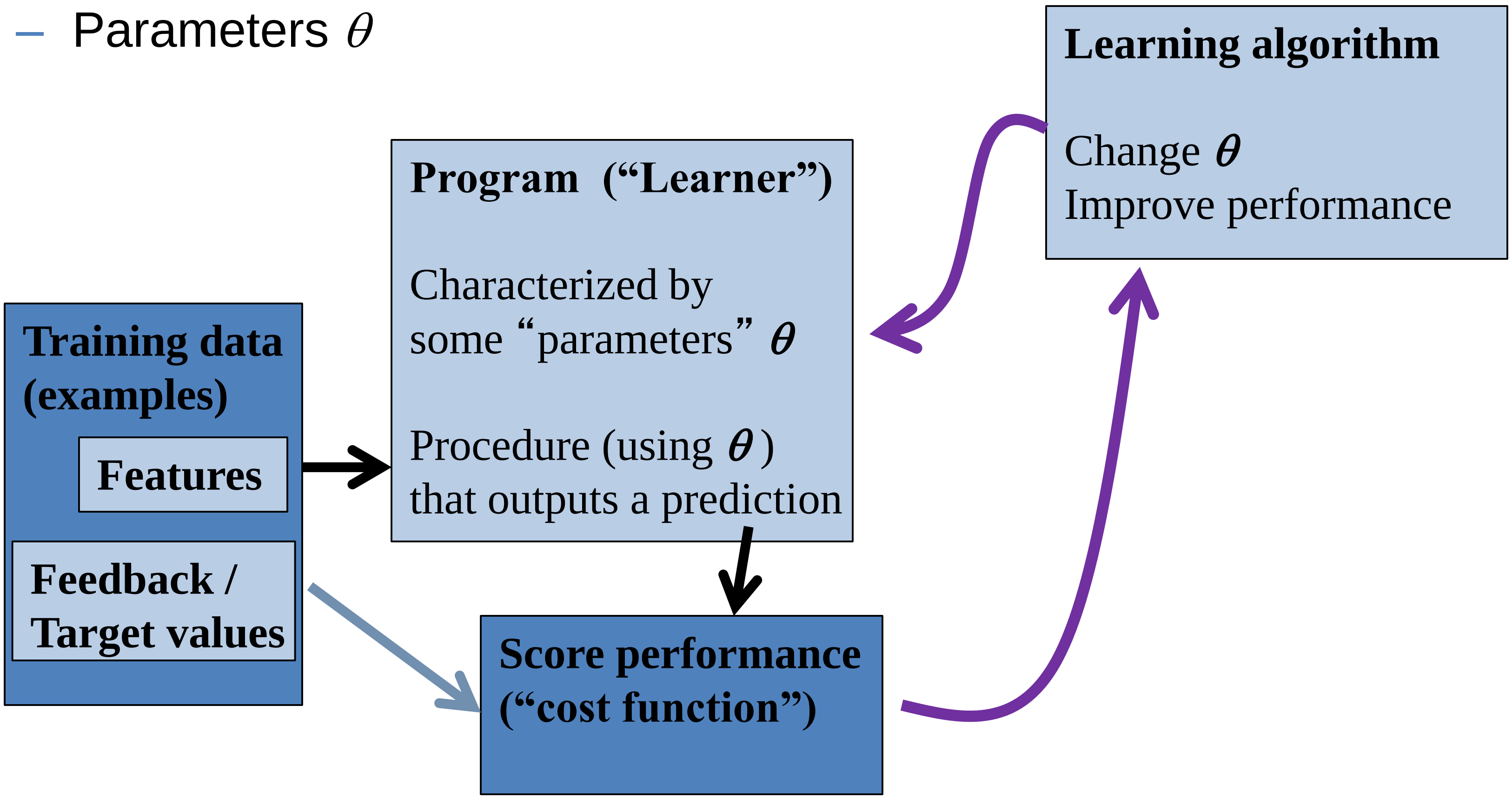
Nearest neighbor methods

Kalev Kask

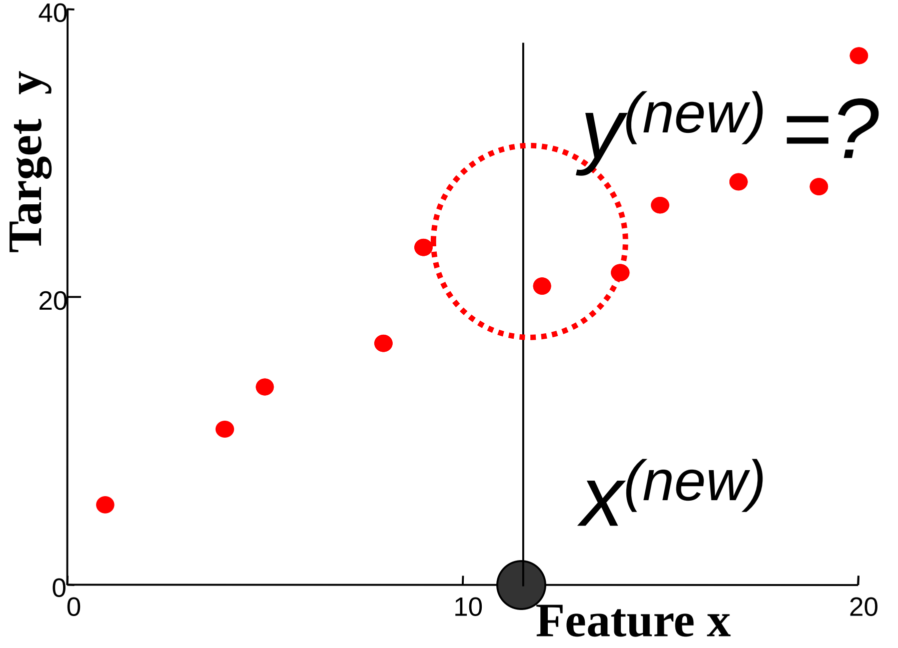


Supervised learning

- Notation
  - Features x
  - Targets y
  - Predictions $\hat{y}$
  - Parameters θ

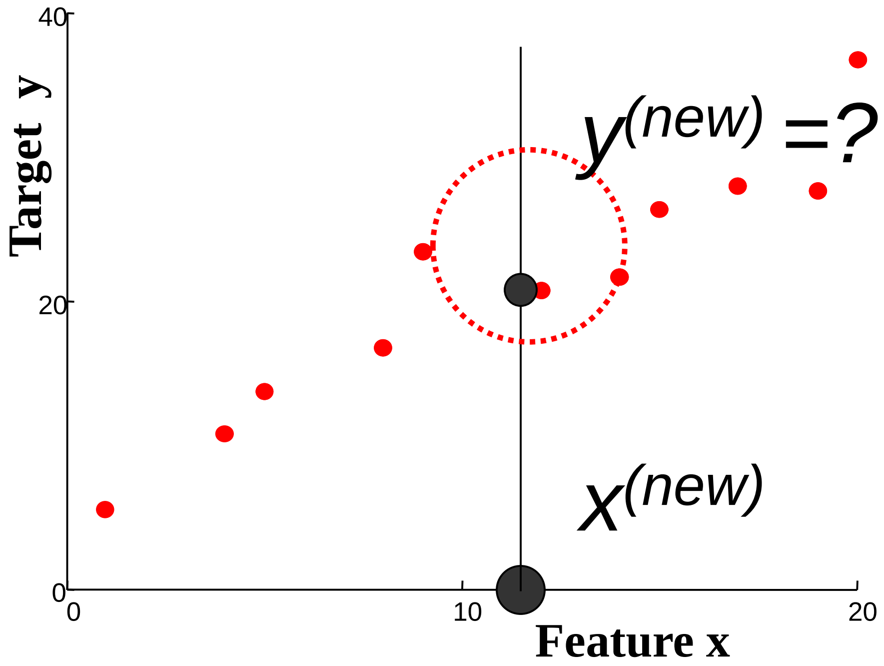


Regression; Scatter plots



- Suggests a relationship between x and y
- Regression: given new observed $x^{(new)}$, estimate $y^{(new)}$

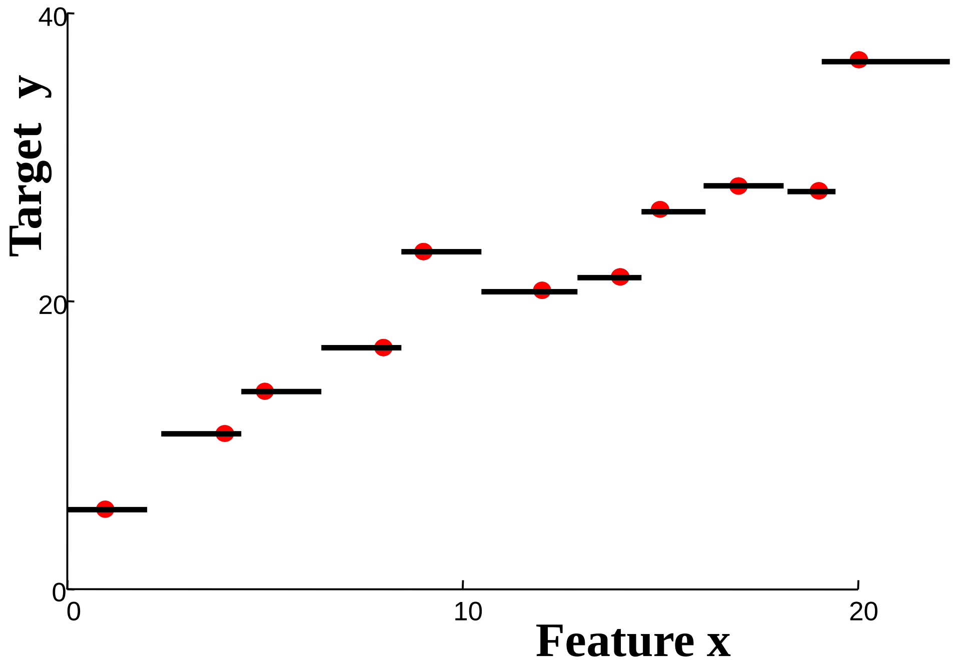
Nearest neighbor regression



"Predictor":
Given new features:
Find nearest example
Return its value

- Find training datum $x^{(i)}$ closest to $x^{(new)}$; predict $y^{(i)}$

Nearest neighbor regression

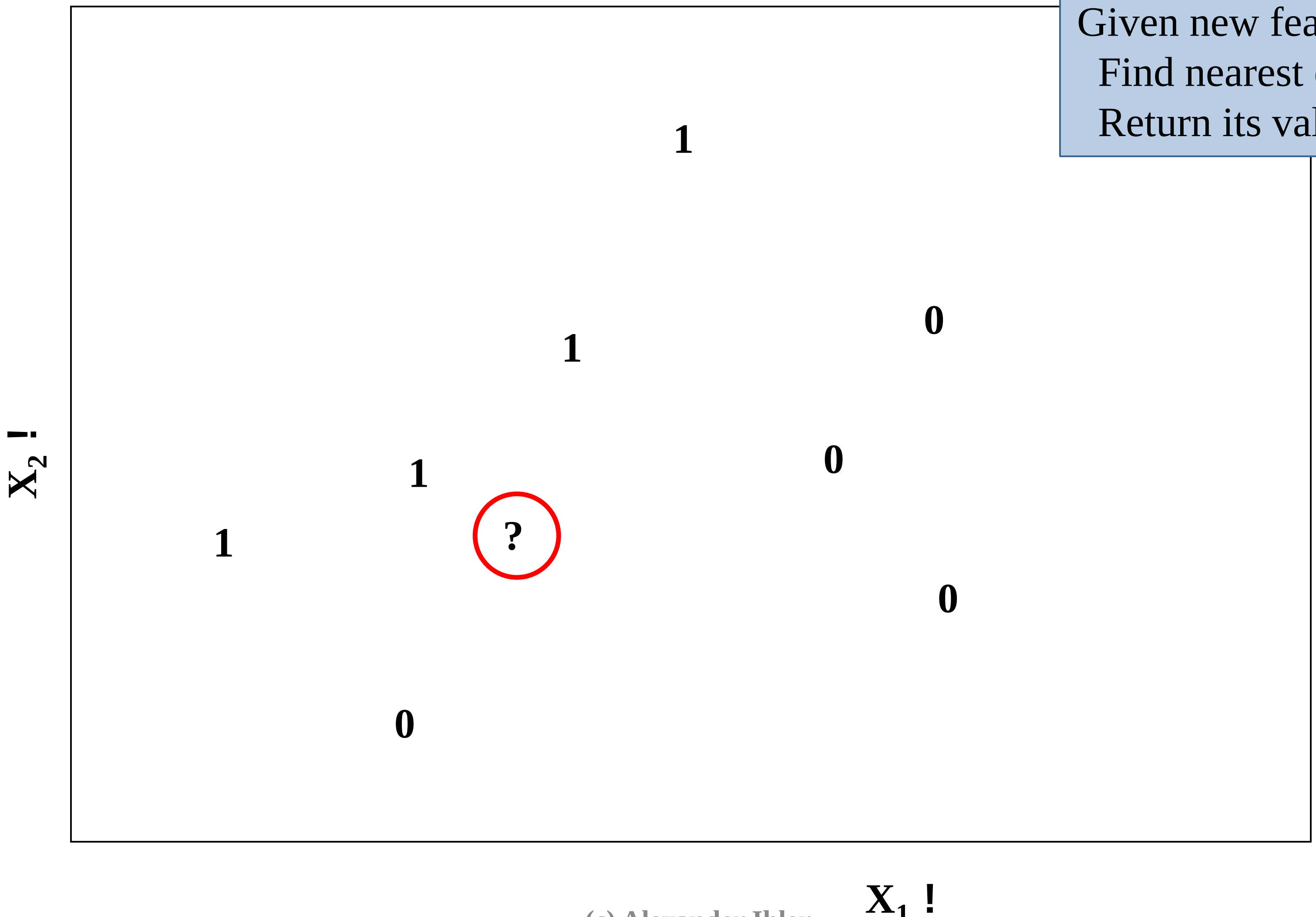


"Predictor":
Given new features:
Find nearest example
Return its value

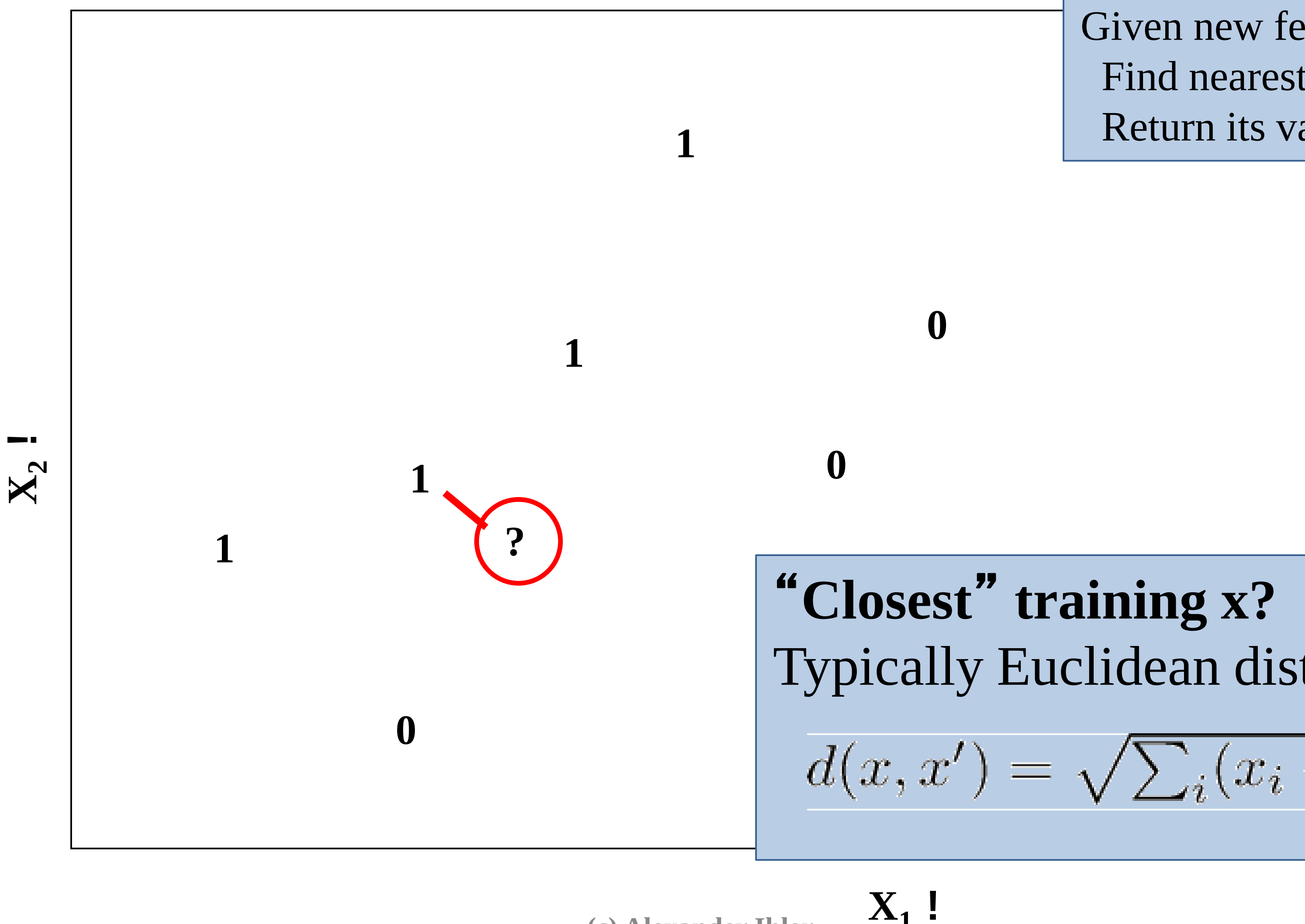
- Find training datum $x^{(i)}$ closest to $x^{(new)}$; predict $y^{(i)}$
- Defines an (implicit) function f(x)
- "Form" is piecewise constant

Nearest neighbor classifier

"Predictor":
Given new features:
  Find nearest example
  Return its value



Nearest neighbor classifier

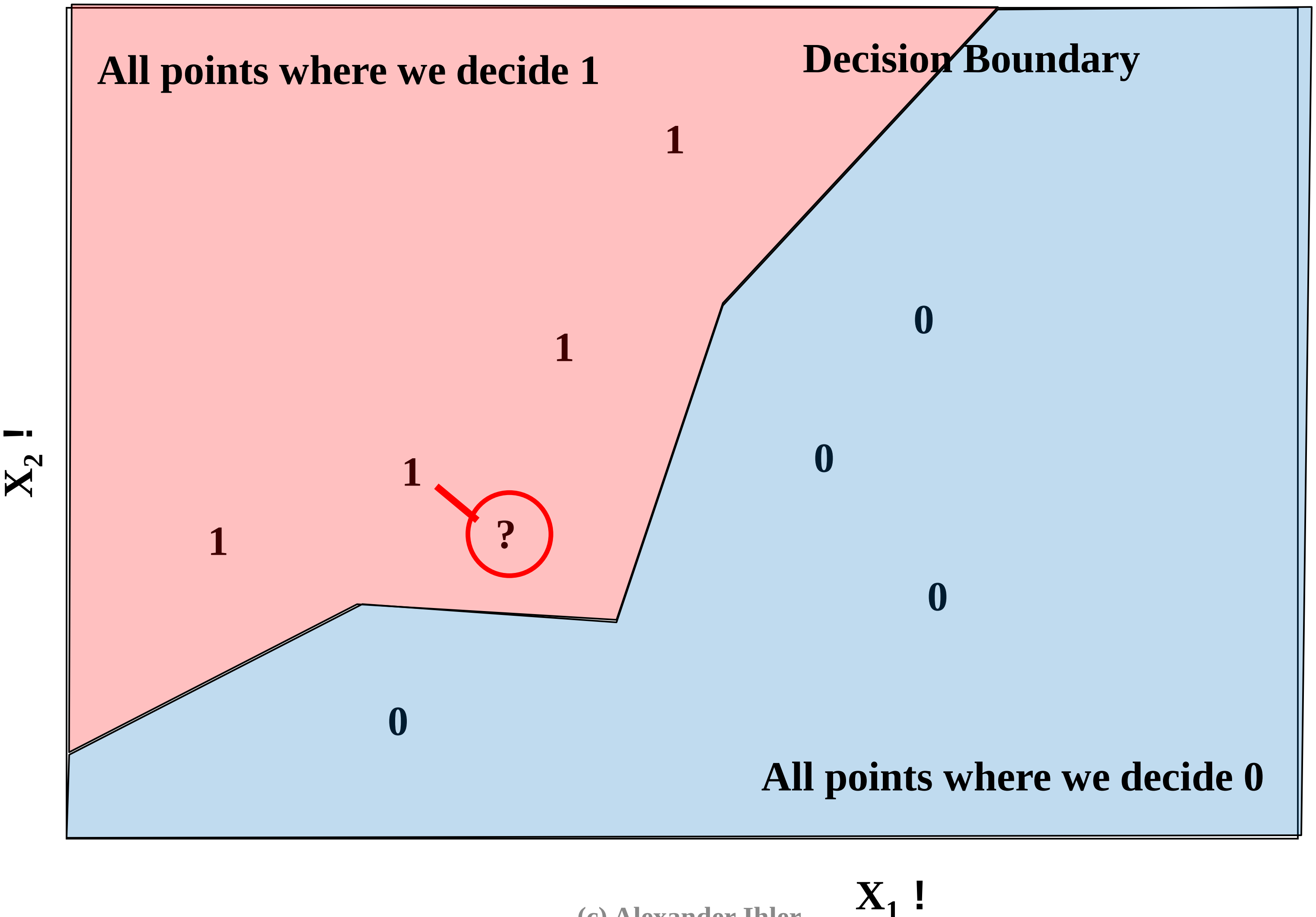


"Predictor":
Given new features:
Find nearest example
Return its value

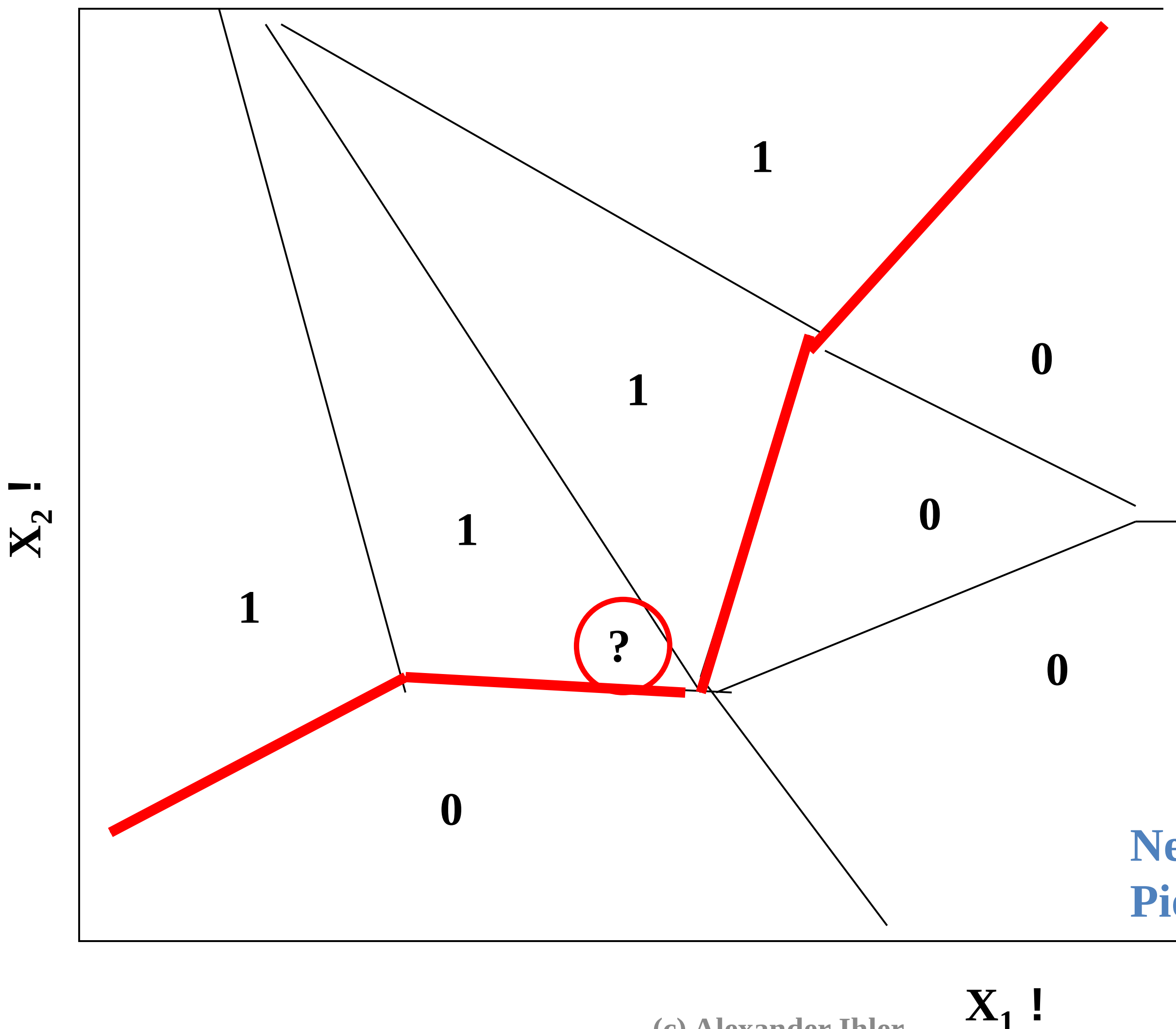
"Closest" training x?
Typically Euclidean distance:

$$d(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$

Nearest neighbor classifier



Nearest neighbor classifier

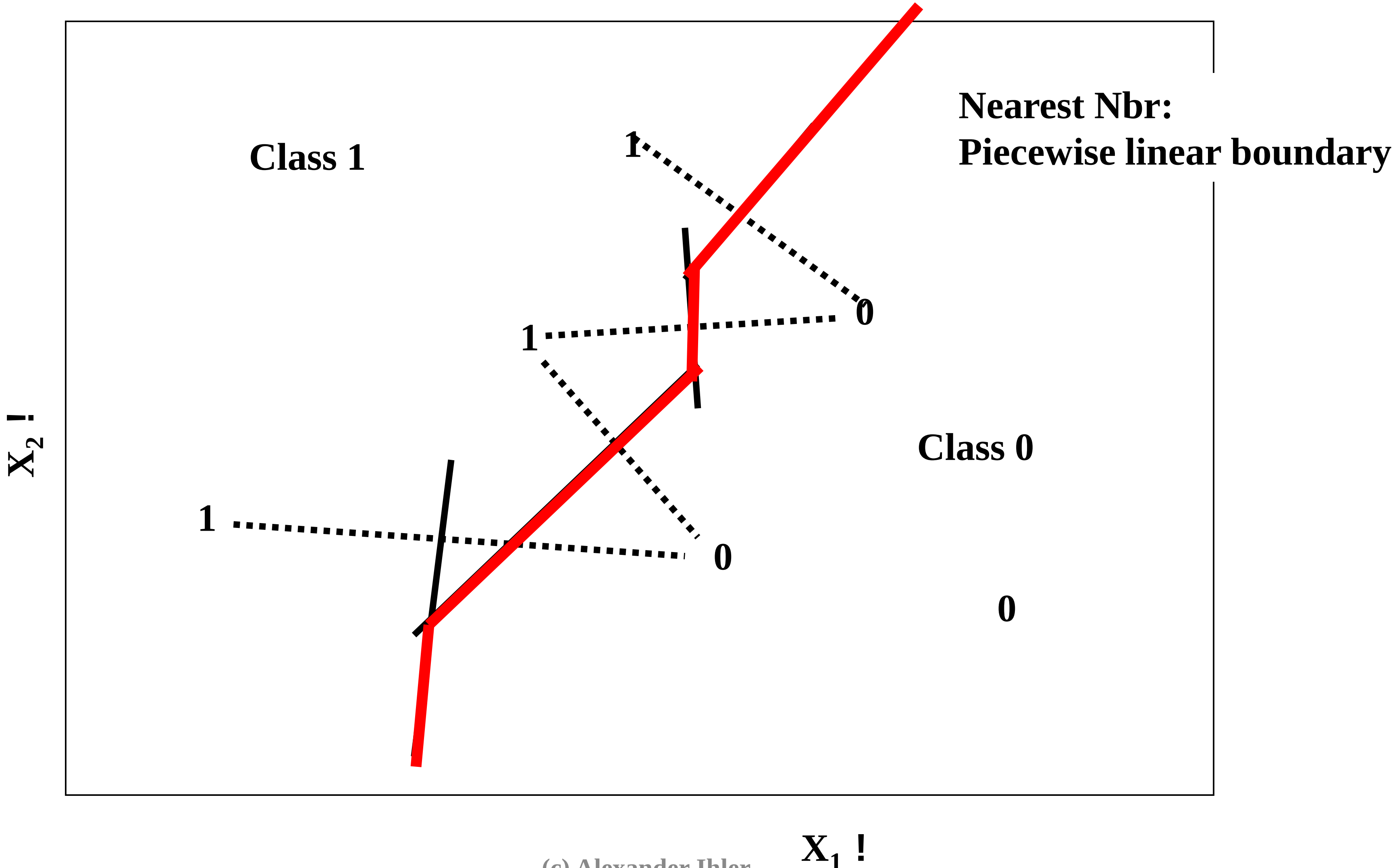


Voronoi tessellation:
Each datum is assigned to a region, in which all points are closer to it than any other datum

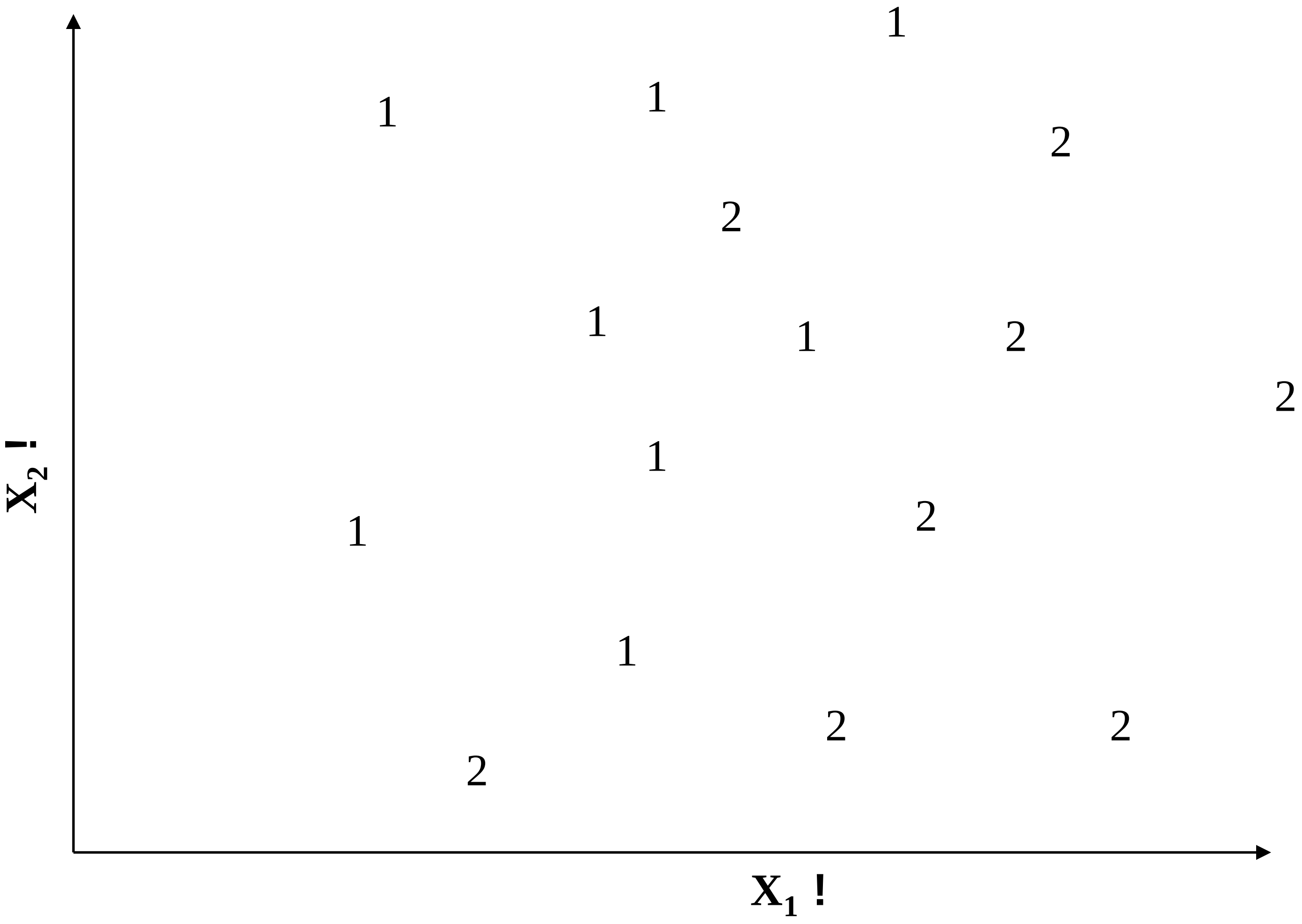
Decision boundary:
Those edges across which the decision (class of nearest training datum) changes

Nearest Nbr:
Piecewise linear boundary

Nearest neighbor classifier

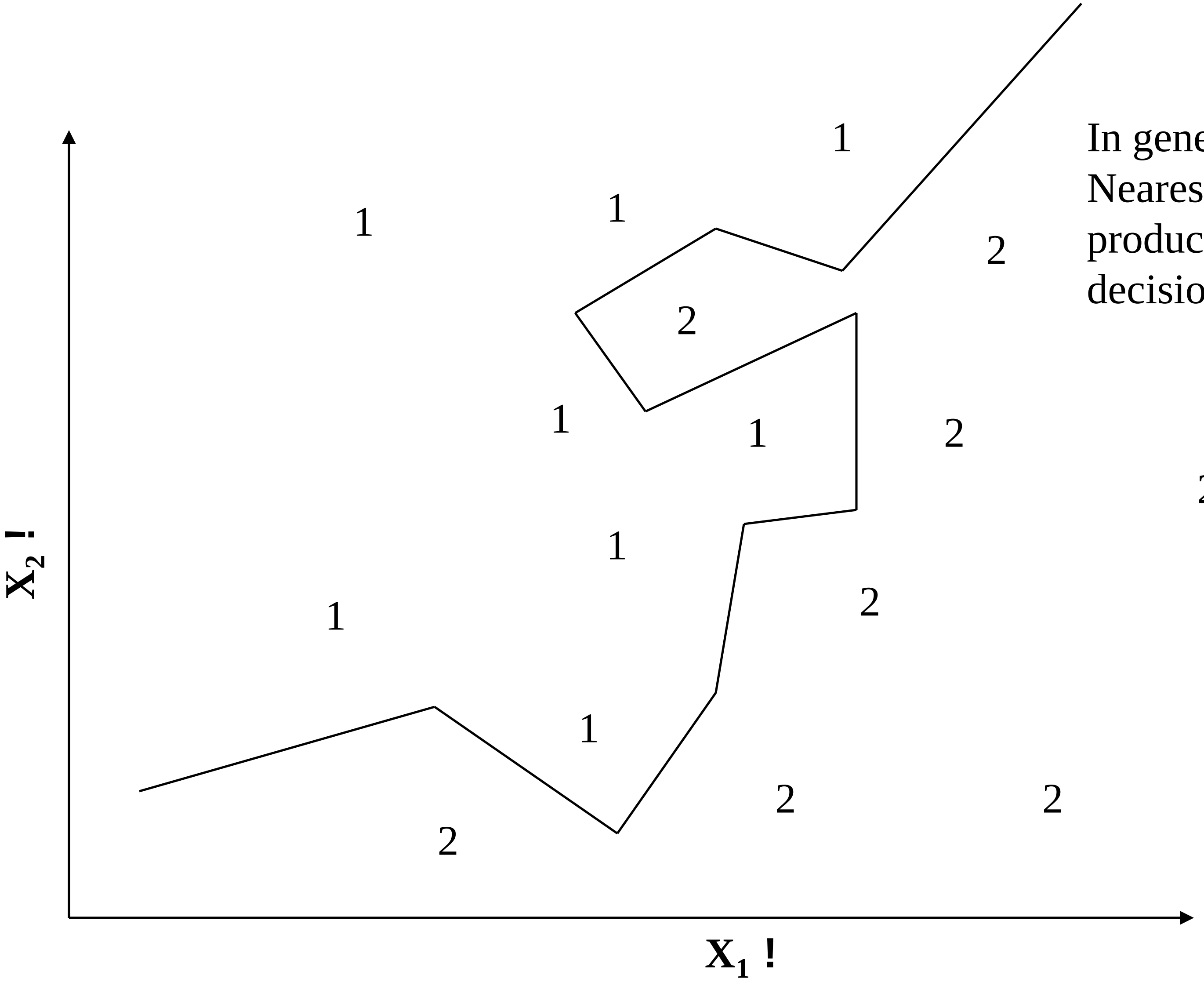


More Data Points



(c) Alexander Ihler

More Complex Decision Boundary



In general:
Nearest-neighbor classifier produces piecewise linear decision boundaries

K-Nearest Neighbor (kNN) Classifier

- Find the k-nearest neighbors to x in the data
  - i.e., rank the feature vectors according to Euclidean distance
  - select the k vectors which are have smallest distance to x

- Regression
  - Usually just average the y-values of the k closest training examples

- Classification
  - ranking yields k feature vectors and a set of k class labels
  - pick the class label which is most common in this set ("vote")
  - classify x as belonging to this class
  - Note: for two-class problems, if k is odd (k=1, 3, 5, …) there will never be any "ties"; otherwise, just use (any) tie-breaking rule

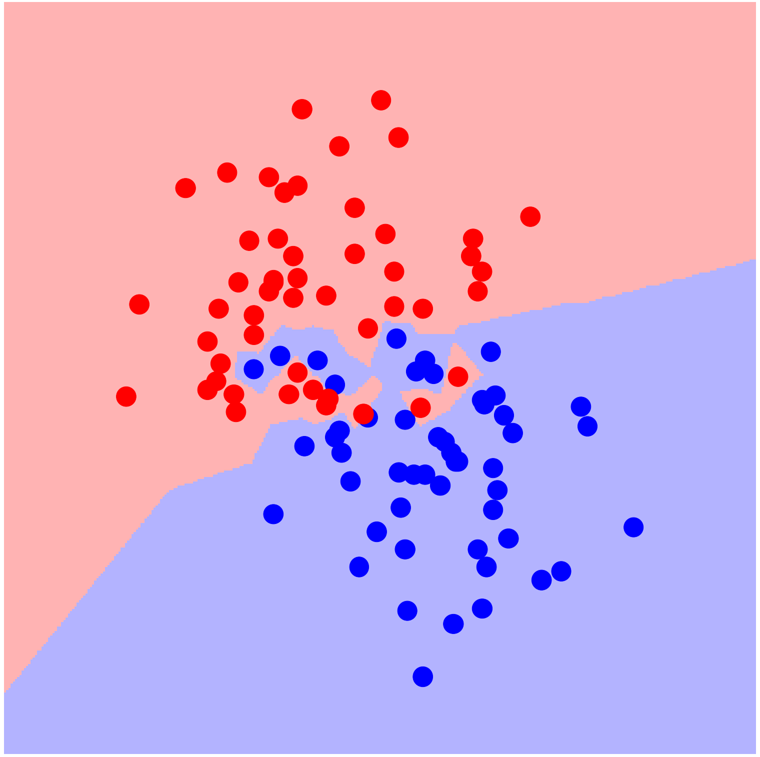
- "Like" the optimal estimator, but using nearest k points to estimate p(y|x)

- "Training" is trivial: just use training data as a lookup table, and search to classify a new datum

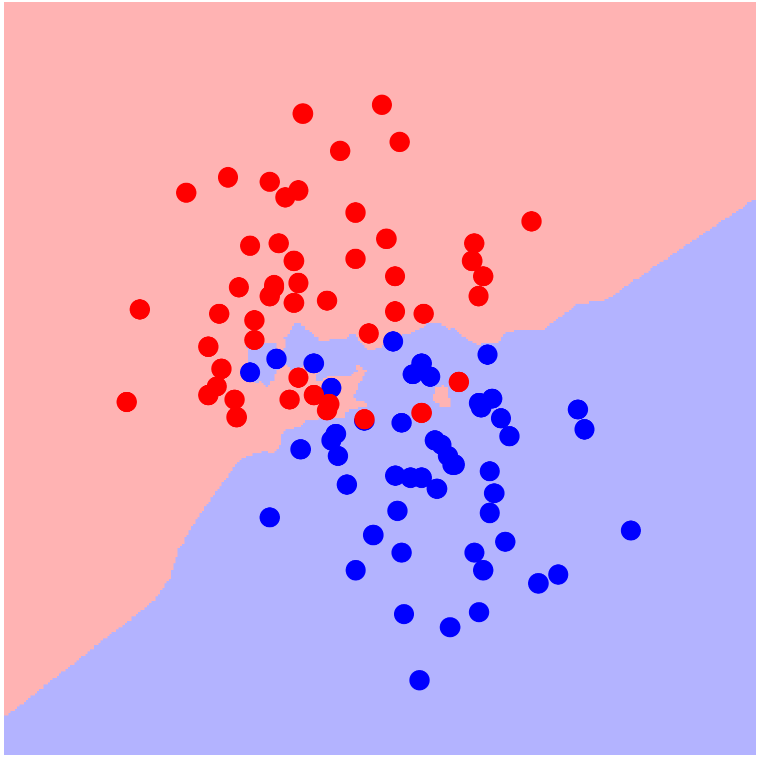
kNN Decision Boundary

- Piecewise linear decision boundary
- Increasing k “simplifies” decision boundary
  - Majority voting means less emphasis on individual points

K = 1



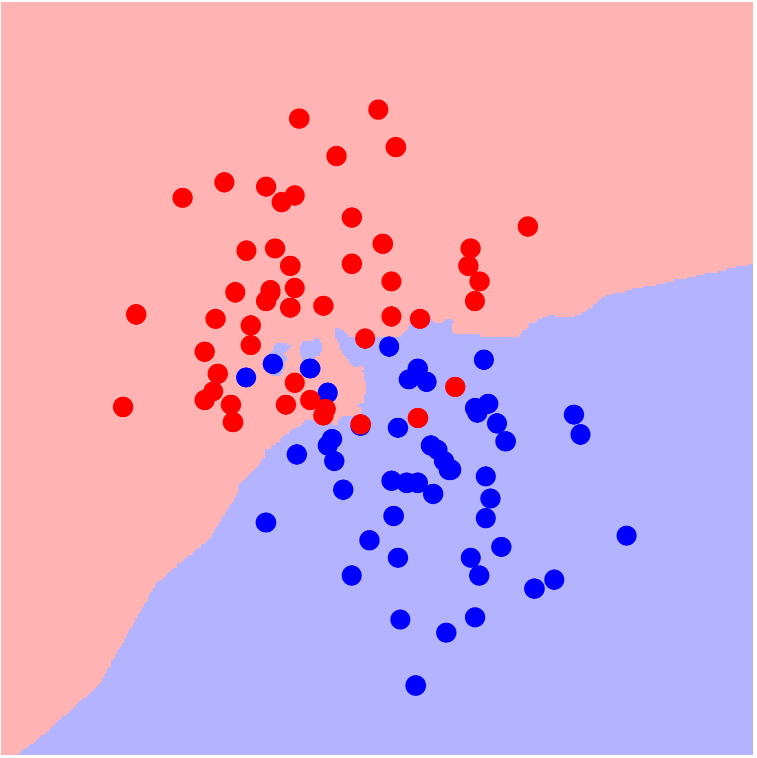
K = 3



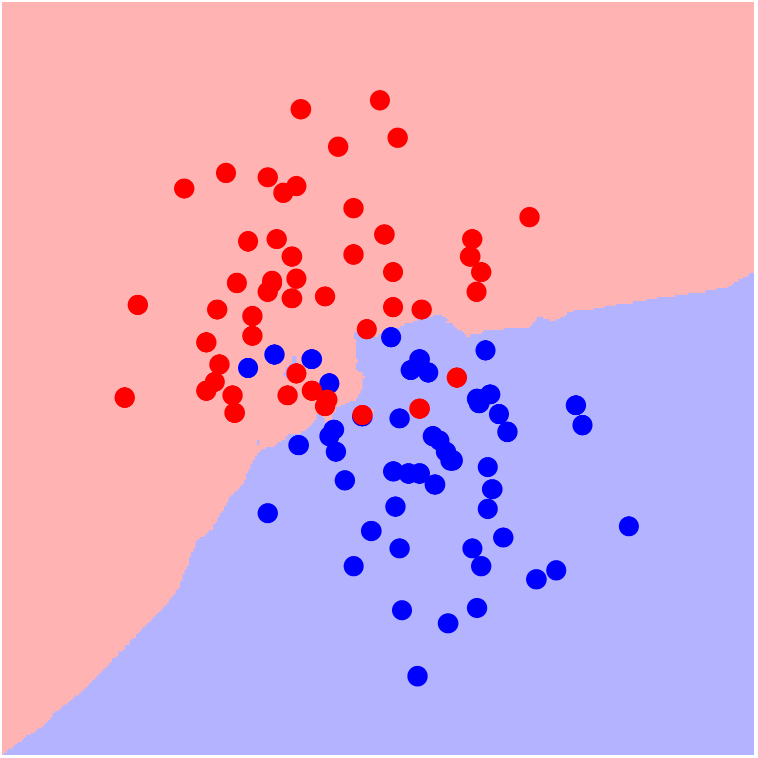
kNN Decision Boundary

- Piecewise linear decision boundary
- Increasing k "simplifies" decision boundary
  - Majority voting means less emphasis on individual points

K = 5



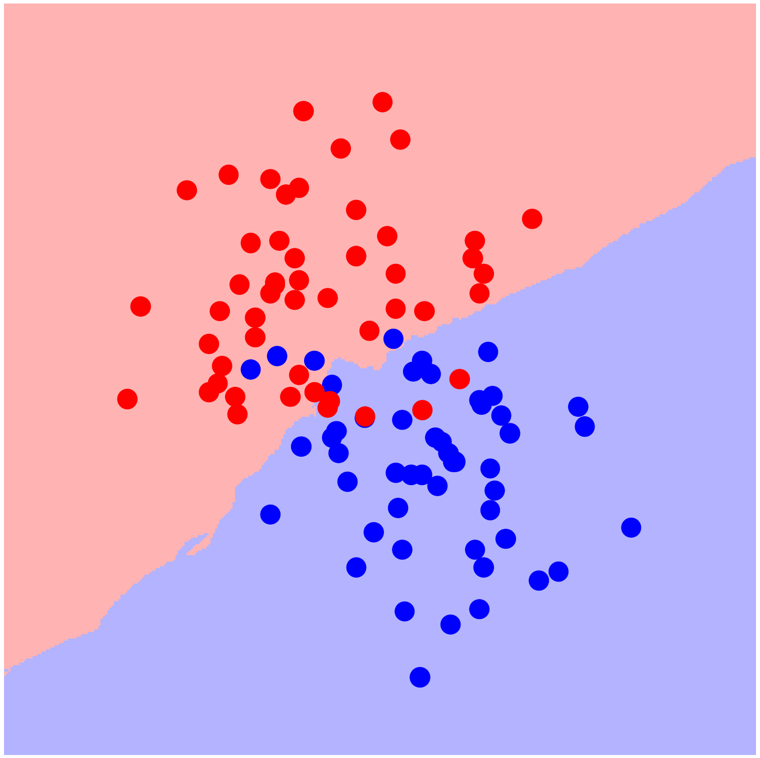
K = 7



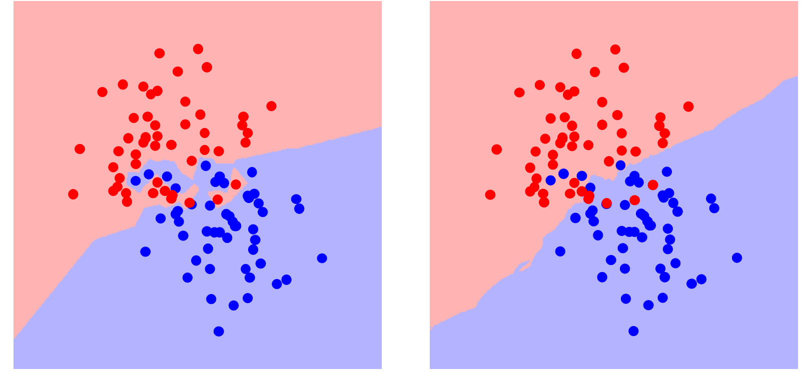
kNN Decision Boundary

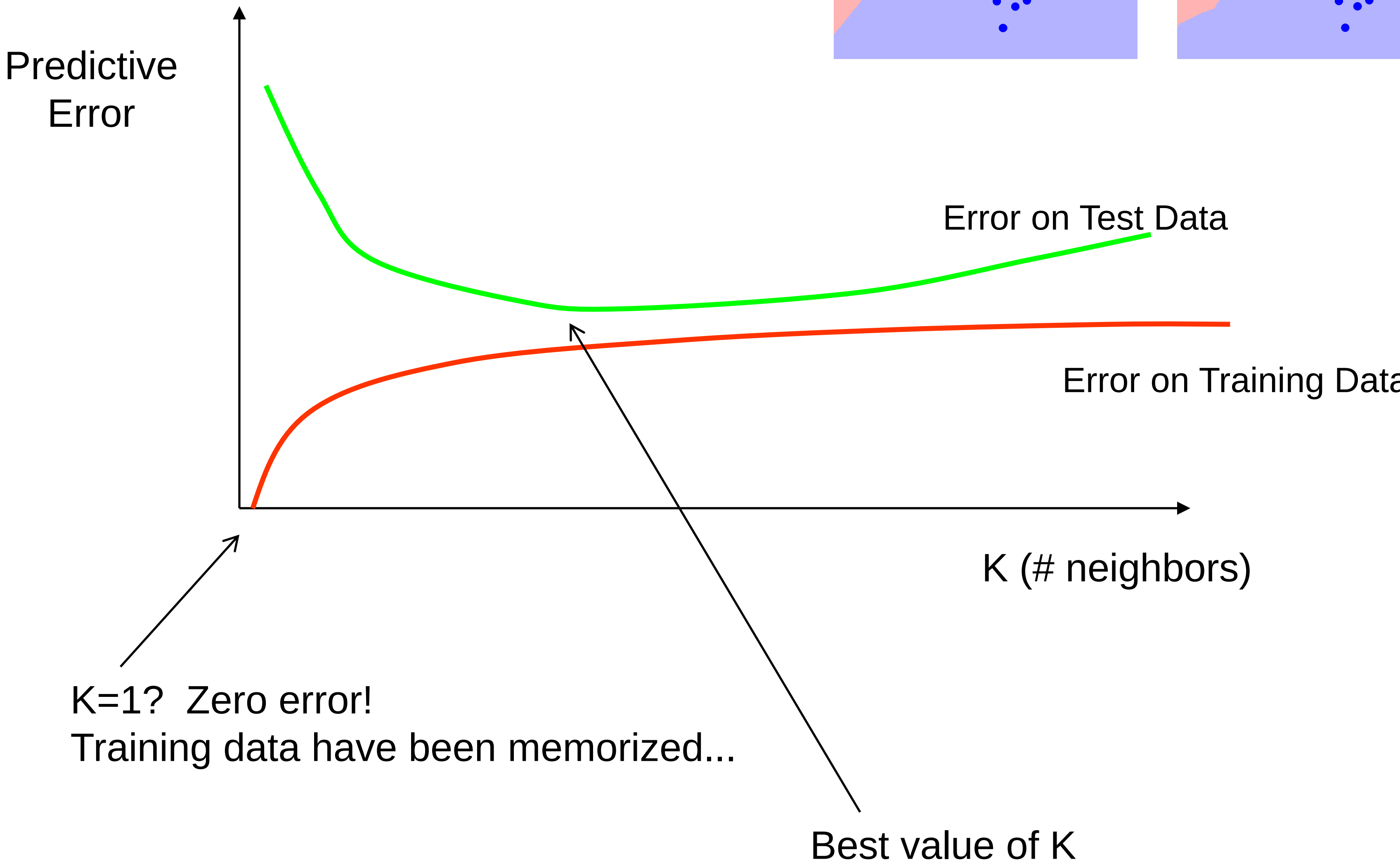
- Piecewise linear decision boundary
- Increasing k “simplifies” decision boundary
  - Majority voting means less emphasis on individual points

K = 25



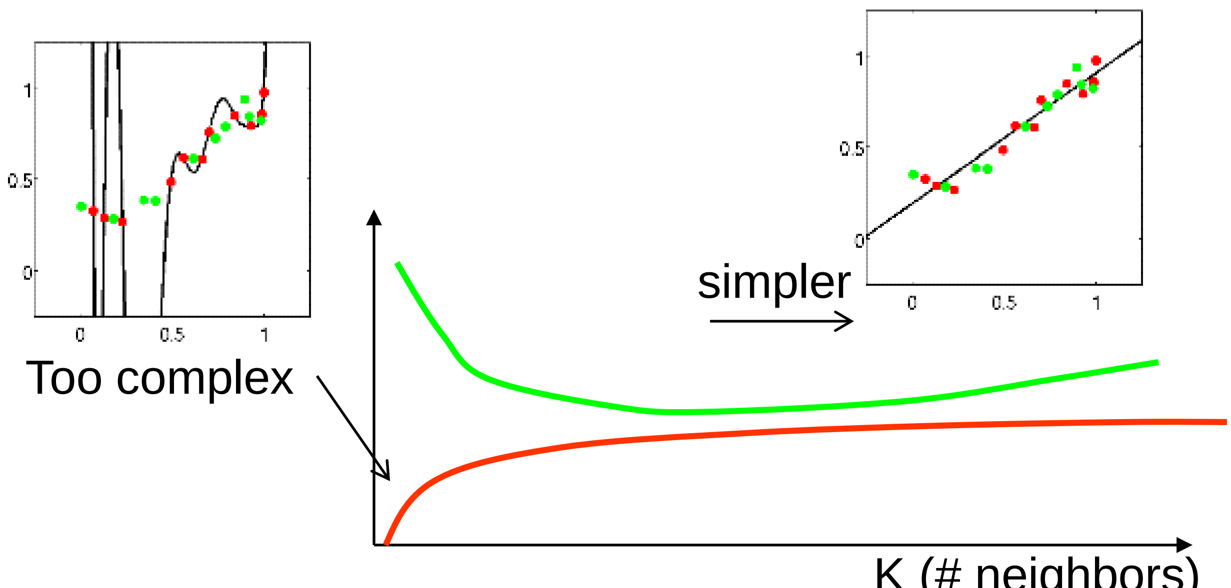
Error rates and K





Complexity & Overfitting

- Complex model predicts all training points well
- Doesn’t generalize to new data points
- k = 1 : perfect memorization of examples (complex)
- k = m : always predict majority class in dataset (simple)
- Can select k using validation data, etc.



K-Nearest Neighbor (kNN) Classifier

- Theoretical Considerations
  - as k increases
    - we are averaging over more neighbors
    - the effective decision boundary is more "smooth"
  - as m increases, the optimal k value tends to increase (as O(log(m)))
  - k=1, m increasing to infinity : error < 2x optimal

- Extensions of the Nearest Neighbor classifier
  - Weighted distances $d(x, x') = \sqrt{\sum_i w_i (x_i - x'_i)^2}$
    - e.g., some features may be more important; others may be irrelevant
    - Mahalanobis distance: $d(x, x') = \sqrt{(x - x') \cdot S^{-1} \cdot (x - x')}$
  - Fast search techniques (indexing) to find k-nearest points in d-space
  - Weighted average / voting based on distance

Curse of dimensionality

- Various phenomena that occur when analyzing and organizing data in higher dimensions (e.g. thousands)
  - When d >> 1 volume of data increases so rapidly that data becomes sparse
  - The amount of data needed for statistical validity grows exponentially with dimensionality
  - E.g. when d >> 1, distances between points become uniform

Summary

- K-nearest neighbor models
  - Classification (vote)
  - Regression (average or weighted average)

- Piecewise linear decision boundary
  - How to calculate

- Test data and overfitting
  - Model “complexity” for knn
  - Use validation data to estimate test error rates & select k