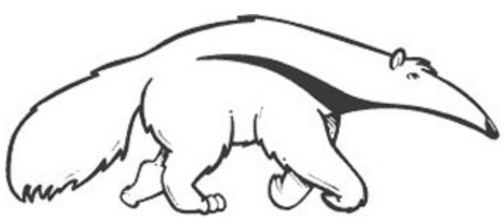
Machine Learning and Data Mining

Introduction

Prof. Alexander Ihler

CS 273a

Fall 2012



BREN:ICS INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Artificial Intelligence (AI)

- CS271
- Building “intelligent systems”
- Lots of parts to intelligent behavior

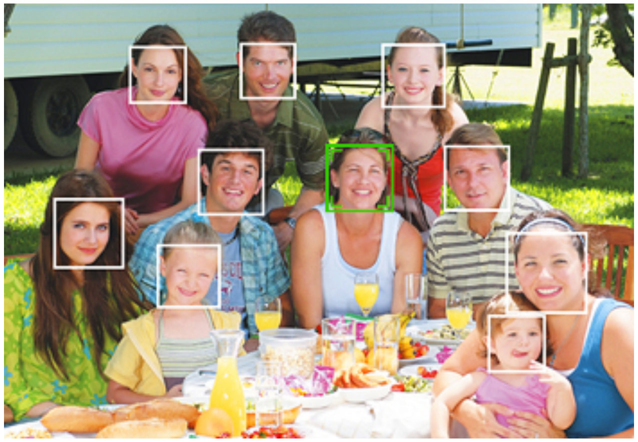
Darpa GC (Stanley)

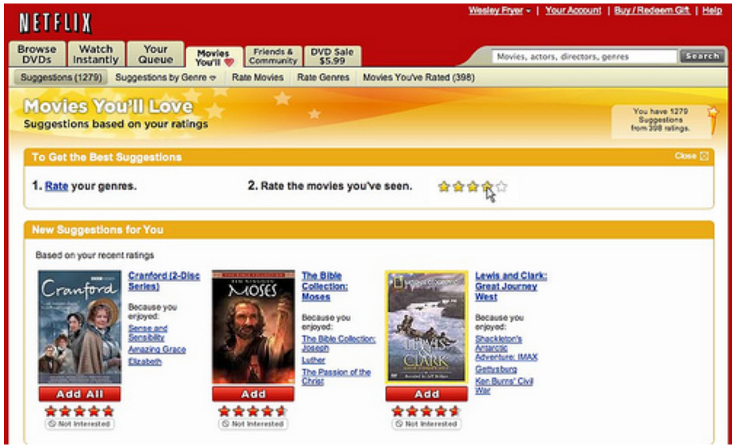
RoboCup

Chess (Deep Blue v. Kasparov)

Machine learning (ML)

- One (important) part of AI
- Making predictions (or decisions)
- Getting better with experience (data)
- Problems whose solutions are “hard to describe”

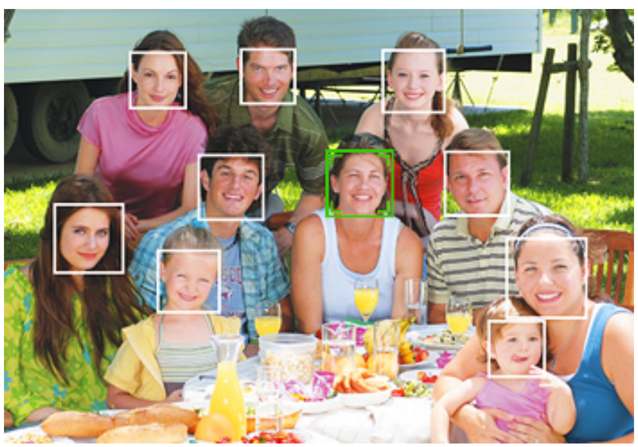


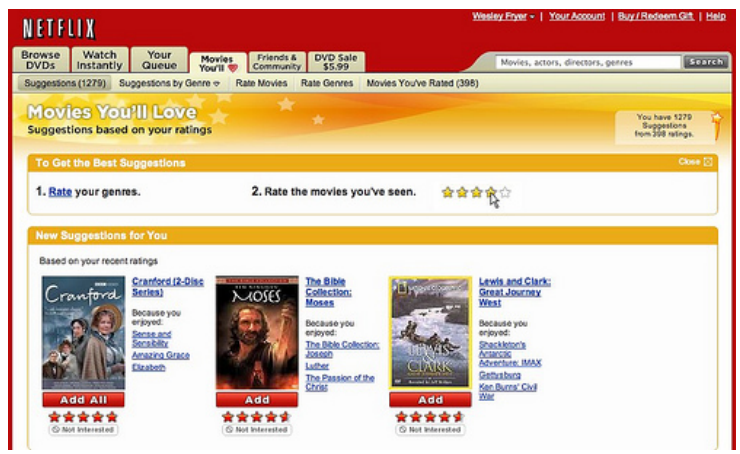


Types of prediction problems

- Supervised learning
  - “Labeled” training data
  - Every example has a desired target value (a “best answer”)
  - Reward prediction being close to target

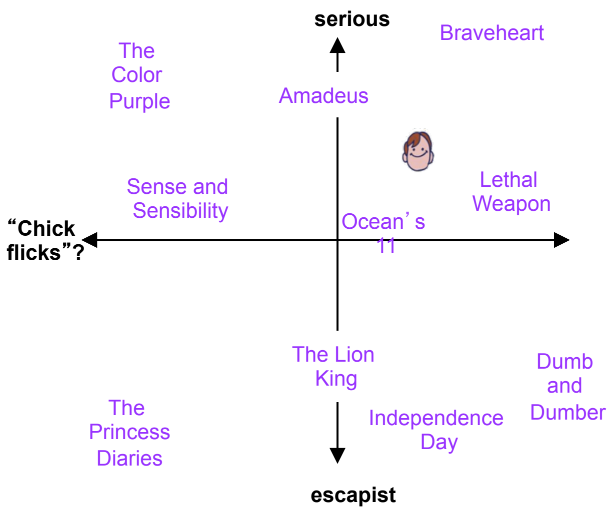
  - Classification: a discrete-valued prediction
  - Regression: a continuous-valued prediction

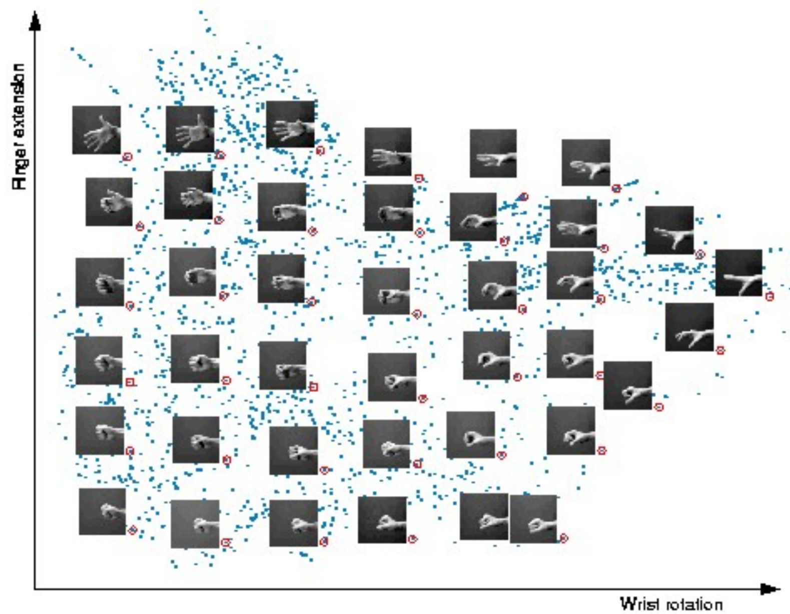




Types of prediction problems

- Supervised learning
- Unsupervised learning
  - No known target values
  - No targets = nothing to predict?
  - Reward “patterns” or “explaining features”
  - Often, data mining





Types of prediction problems

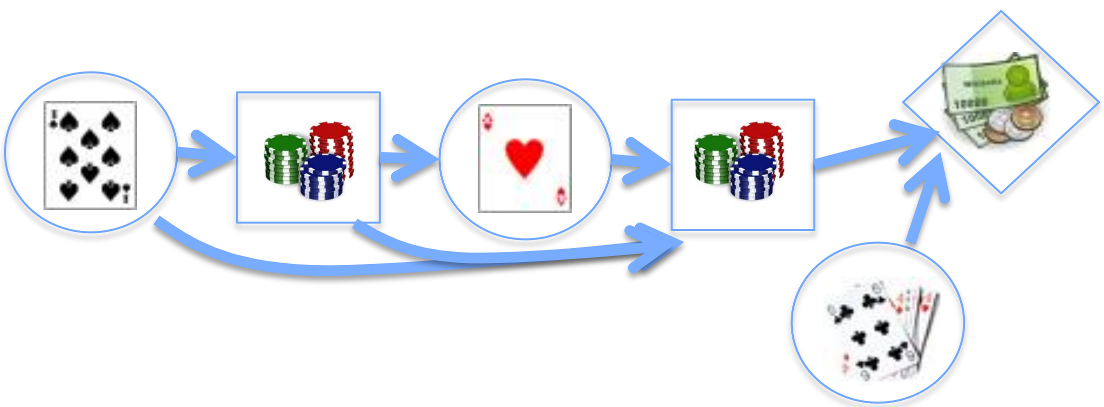
- Supervised learning
- Unsupervised learning
- Semi-supervised learning
  - Similar to supervised
  - some data have unknown target values

- Ex: medical data
  - Lots of patient data, few known outcomes

Types of prediction problems

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

- “Indirect” feedback on quality
  - No answers, just “better” or “worse”
  - Feedback may be delayed



Logistics

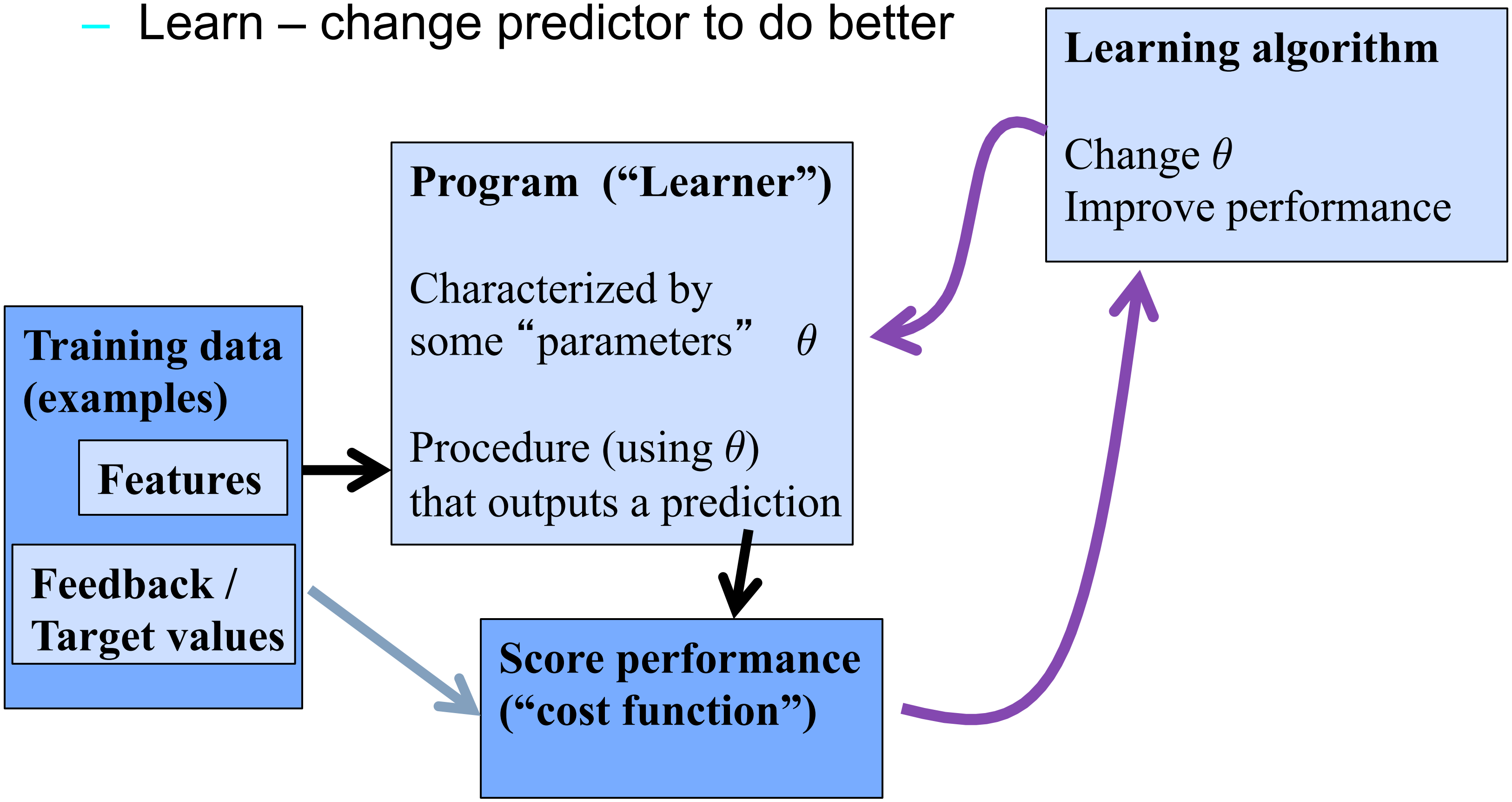
- Course webpage for assignments, info, comments
- EEE for homework, &c
  - Emails: will send a test email tomorrow – make sure you get it
- No required textbook
  - Highly recommended: Murphy, “Machine Learning...”, 2012.
  - Also
    - Duda, Hart & Stork, “Pattern classification”
    - Hastie, Tibshirani & Friedman, “Elements of Statistical Learning”
- But
  - I’ll try to cover everything needed in lectures and notes
  - All textbooks mainly for reference purposes

Logistics

- Grading (approximate)
  - 20% homework (~5-6, drop lowest)
  - 15% project (Kaggle HHC)
  - 5% reading quizzes
  - 25% midterm, 35% final
  - Due 5pm listed day, EEE or my office
  - No late homework (solutions posted)
    - Turn in what you have
- Collaboration
  - Study groups, discussion, assistance encouraged
    - Whiteboards, etc.
  - Do your homework yourself
    - Don't exchange solutions or HW code

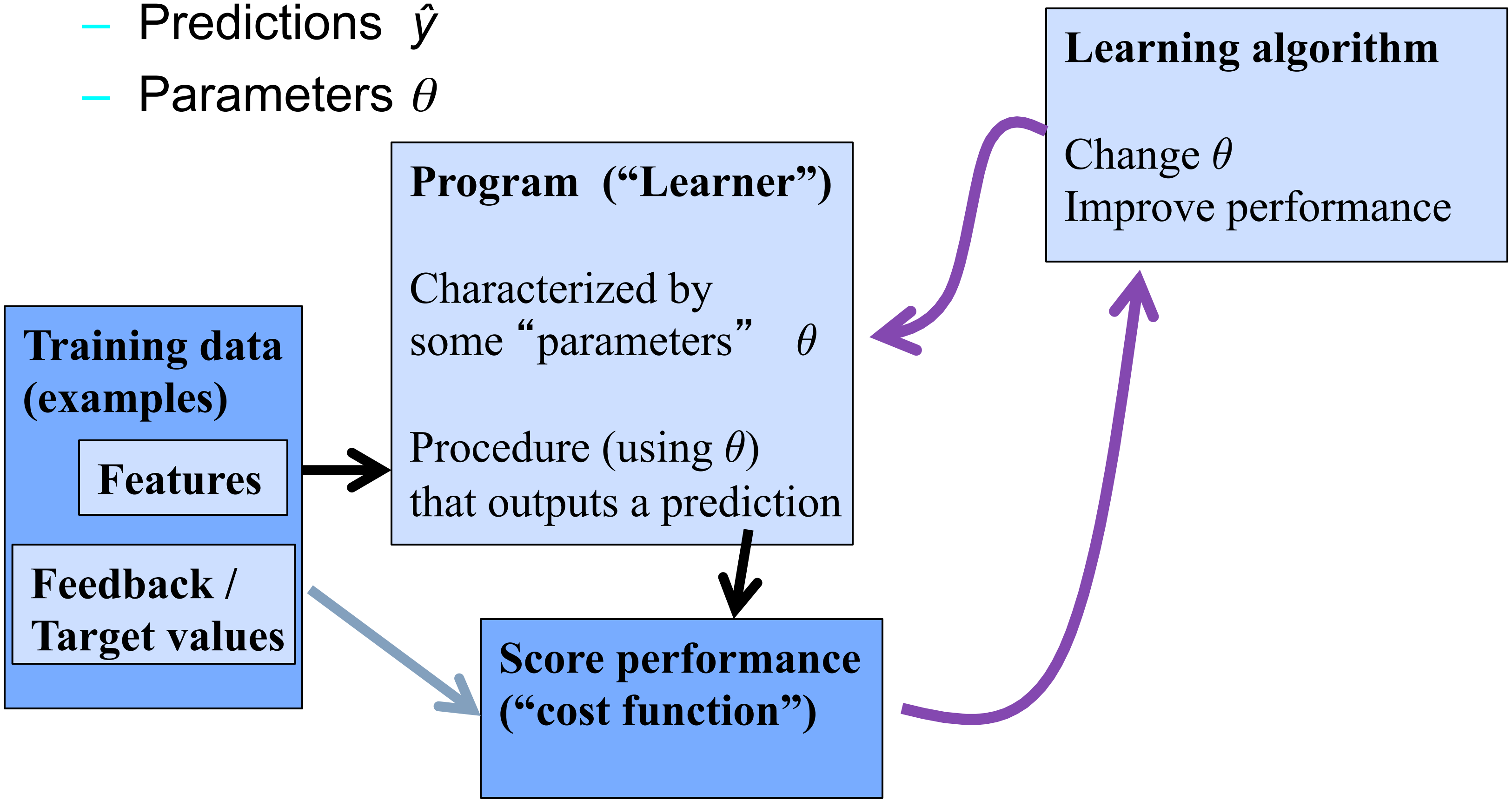
How does machine learning work?

- "Meta-programming"
  - Predict – apply rules to examples
  - Score – get feedback on performance
  - Learn – change predictor to do better

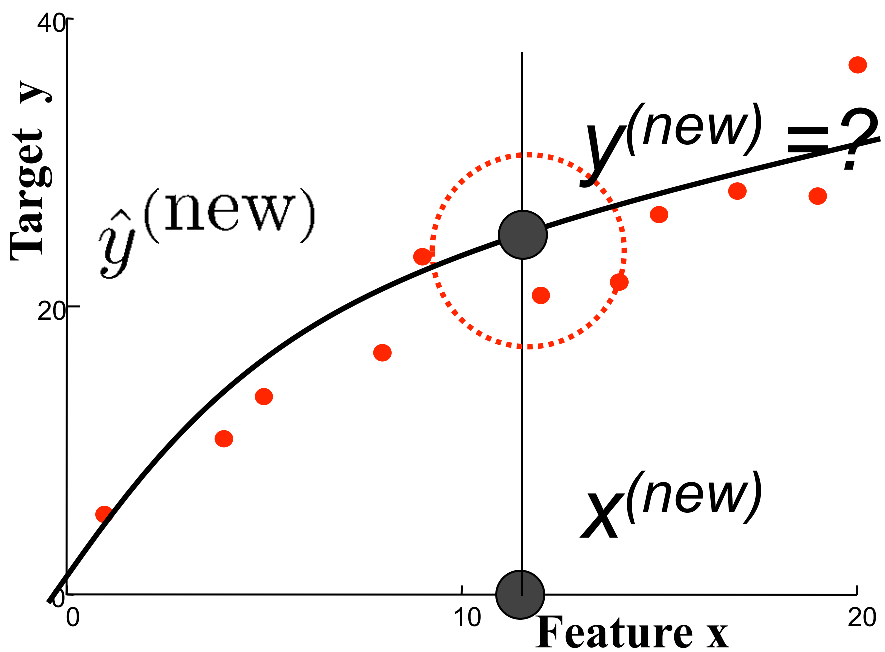


Supervised learning

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

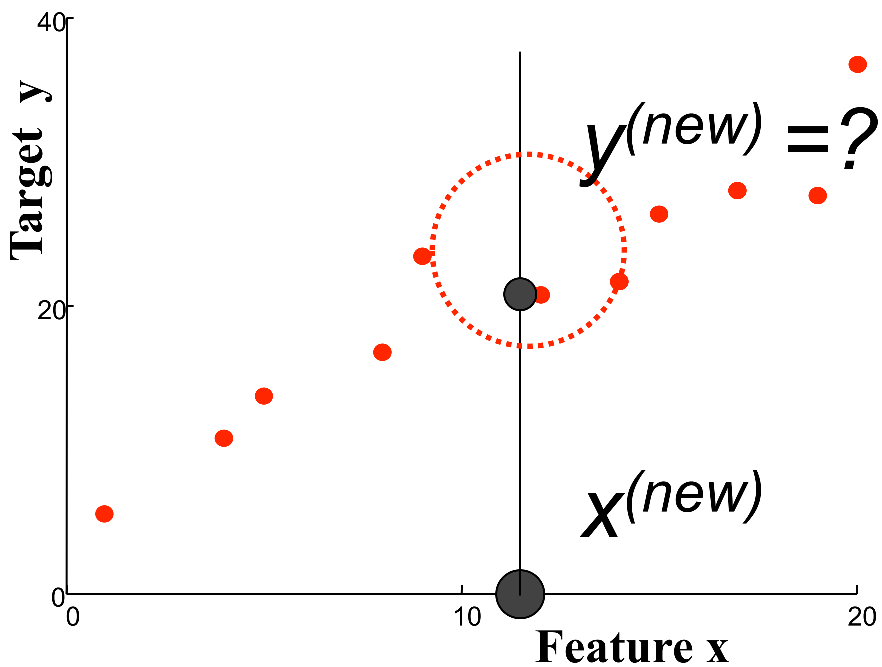


Regression; Scatter plots



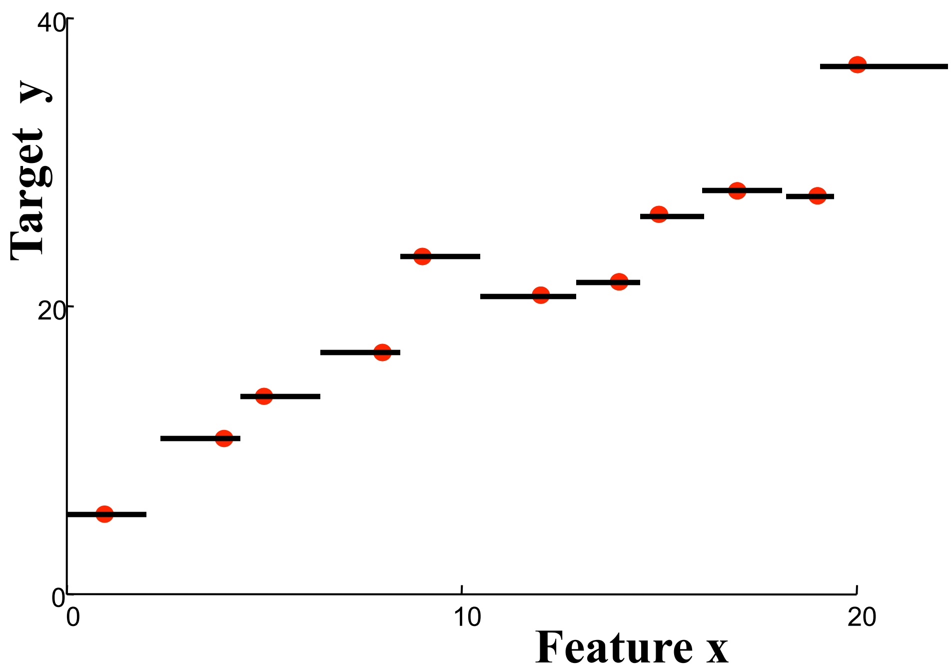
- Suggests a relationship between x and y
- *Prediction*: new x, what is y?

Nearest neighbor regression



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

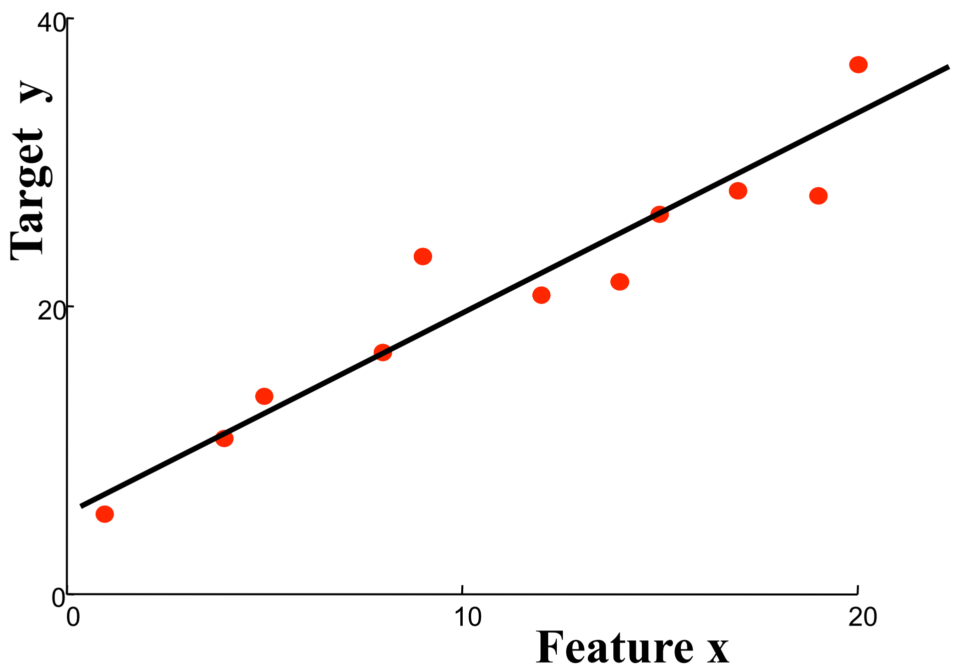
Nearest neighbor regression



“Predictor”:
Given new features:
  Find nearest example
  Return its value

- Defines a function f(x) implicitly
- “Form” is piecewise constant

Linear regression



"Predictor":
Evaluate line:

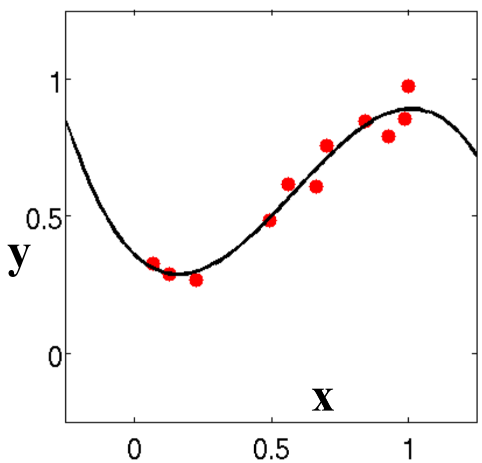
$$r = \theta_0 + \theta_1 x_1$$

return r

- Define form of function f(x) explicitly
- Find a good f(x) within that family

Regression vs. Classification

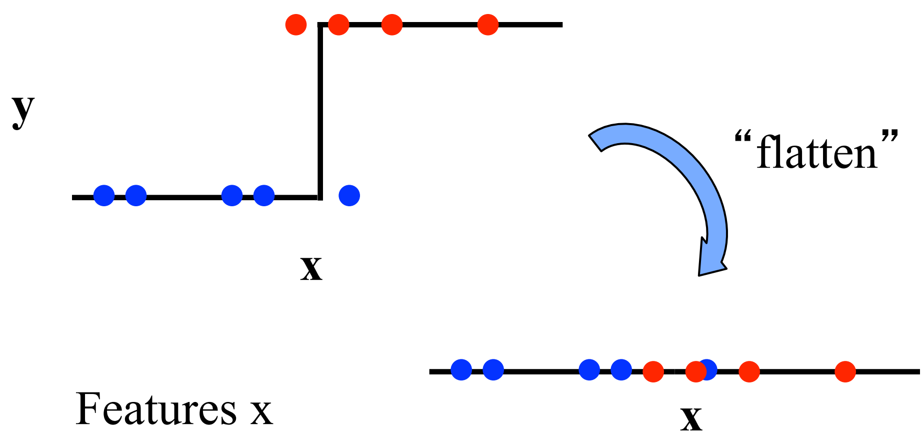
Regression



Features x
Real-valued target y

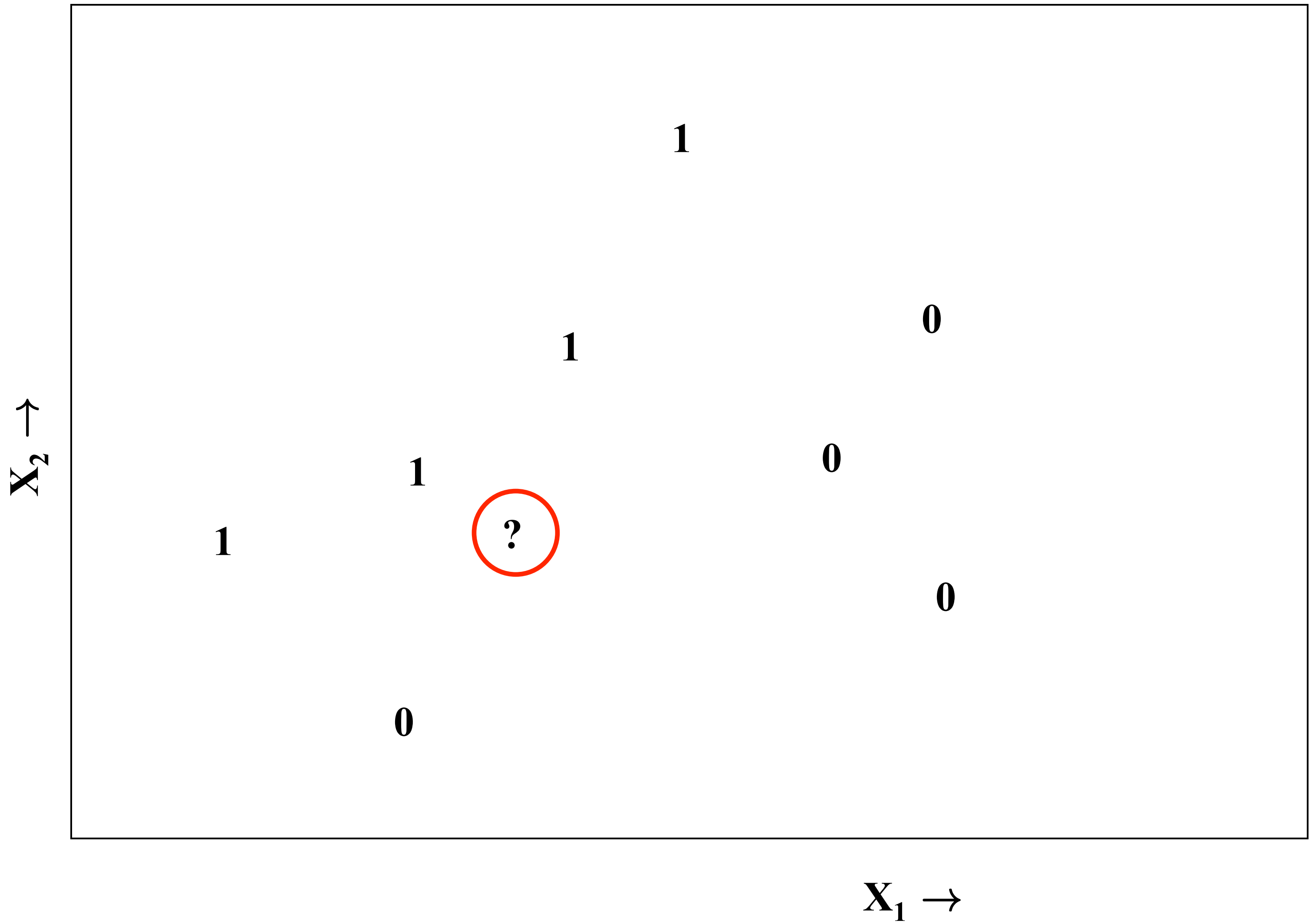
Predict continuous function ŷ(x)

Classification

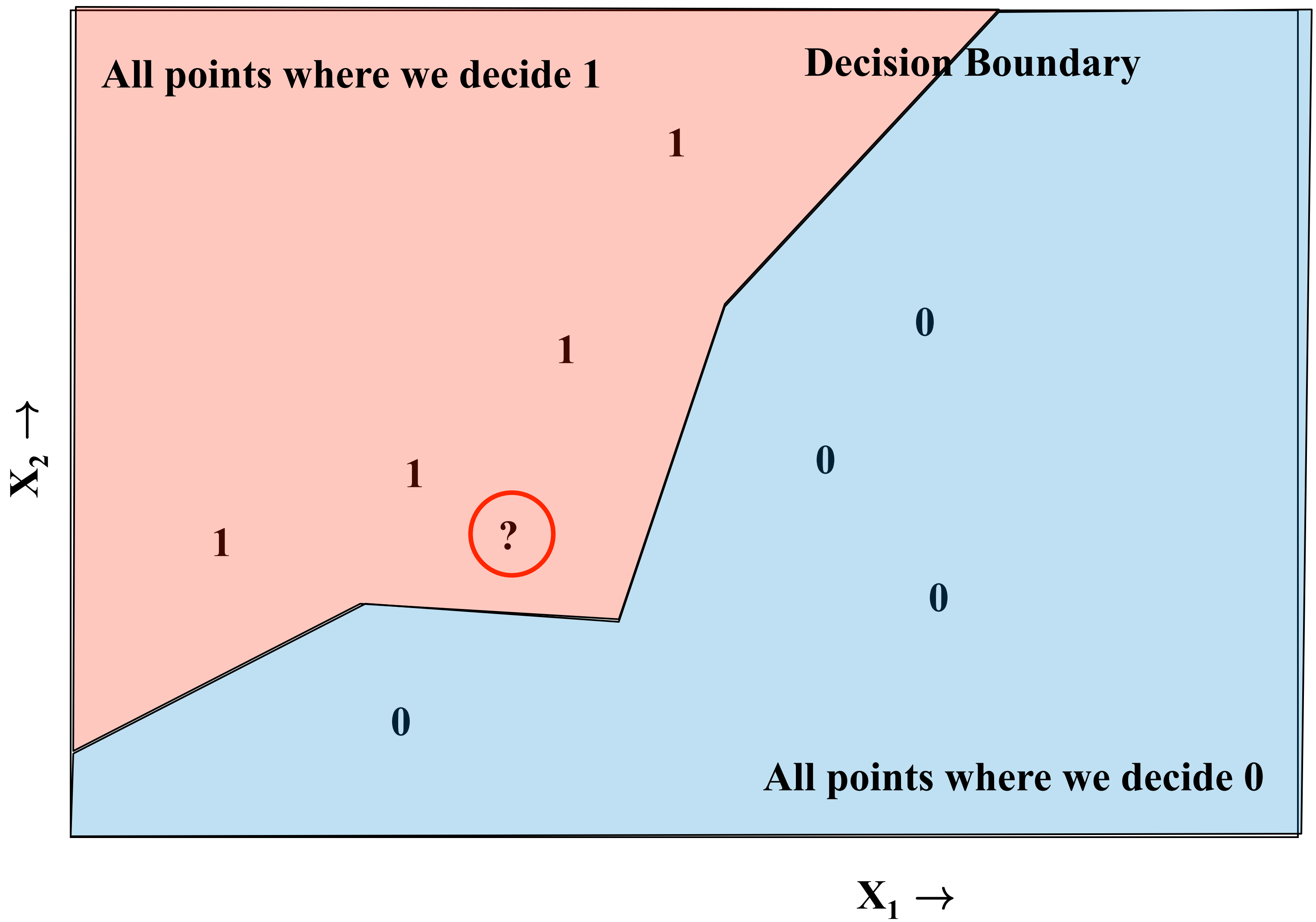


Features x
Discrete class c
(usually 0/1 or +1/-1)
Predict discrete function ŷ(x)

Classification

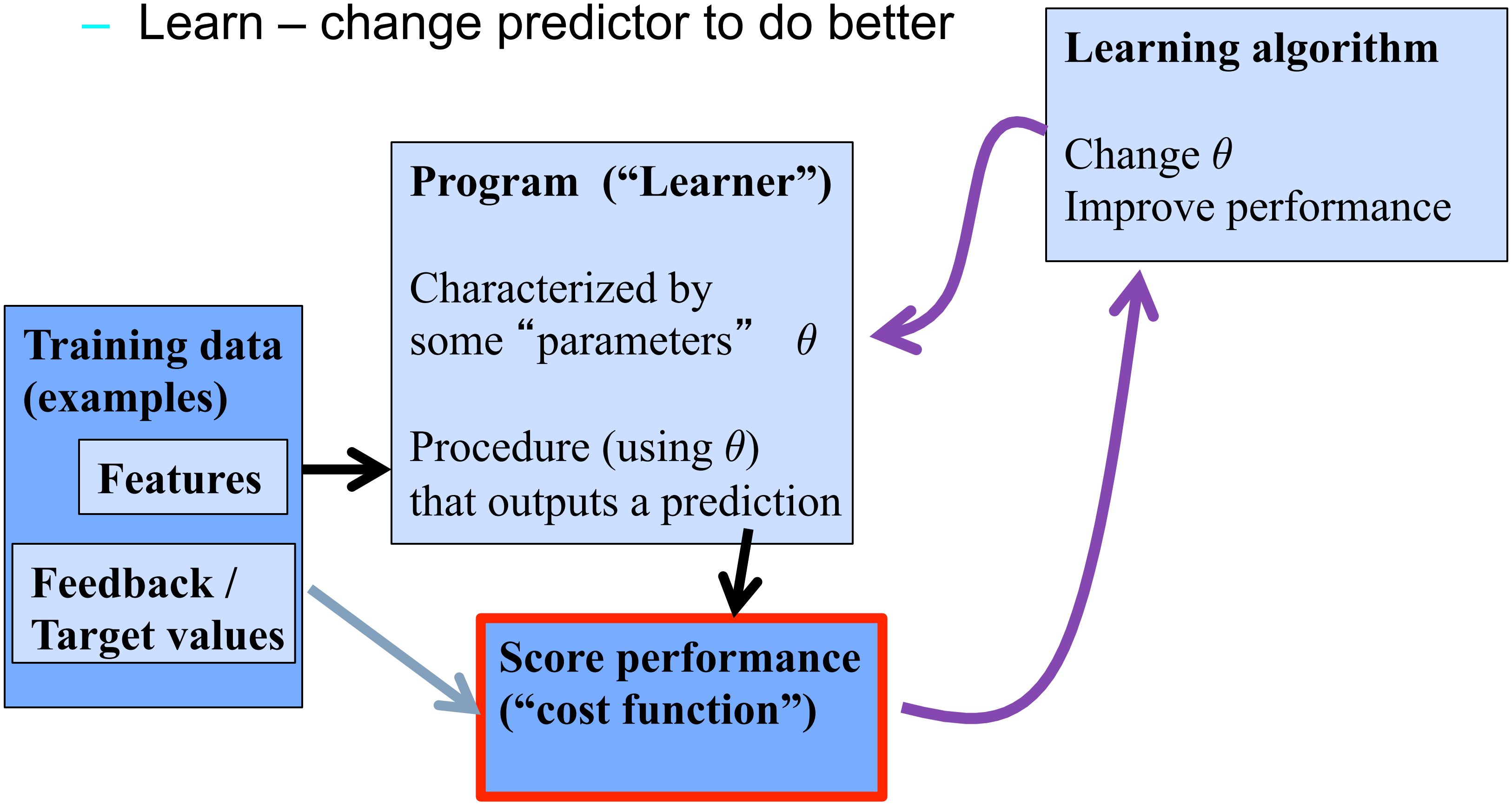


Classification

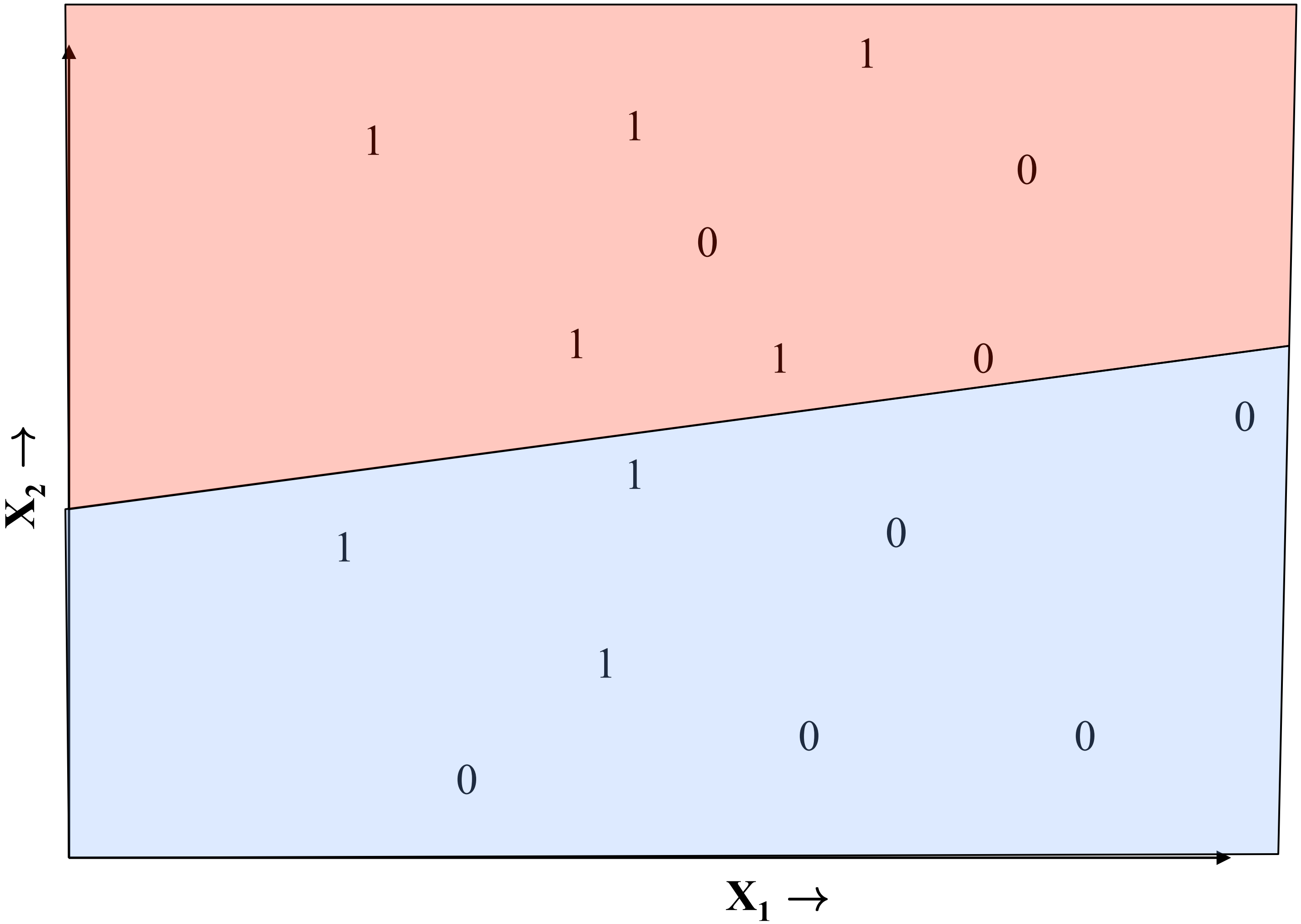


How does machine learning work?

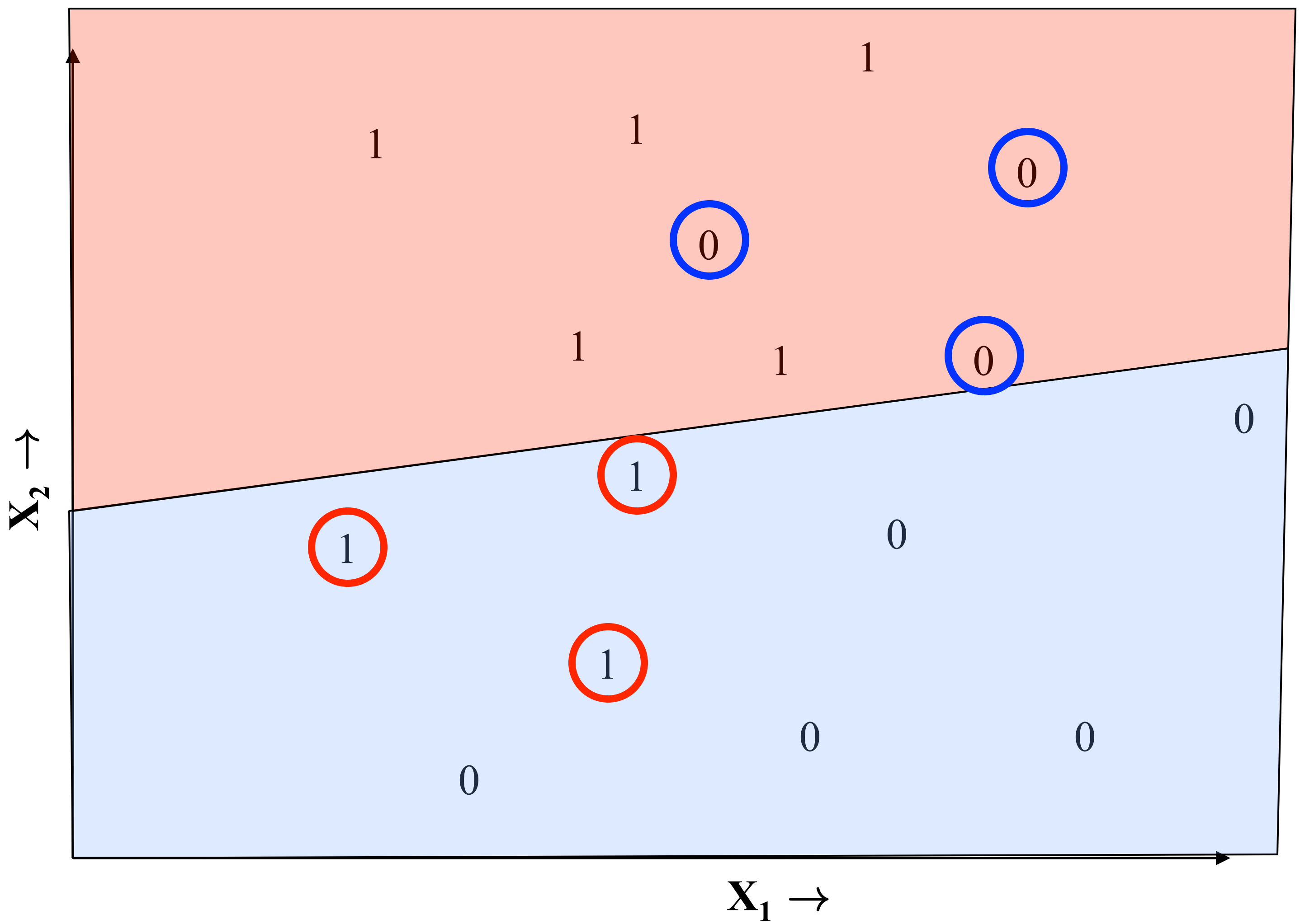
- "Meta-programming"
  - Predict – apply rules to examples
  - Score – get feedback on performance
  - Learn – change predictor to do better



Measuring error

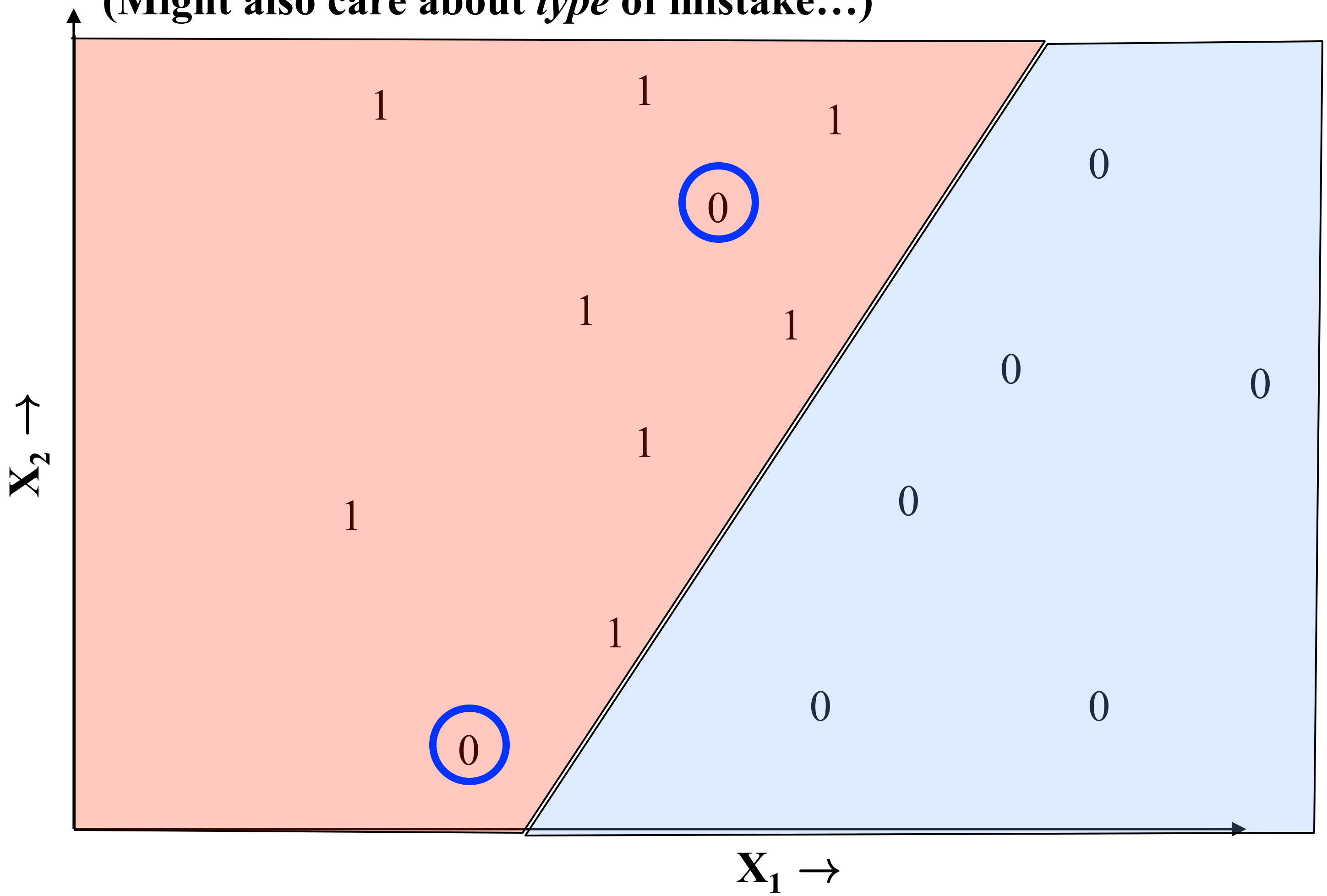


Measuring error

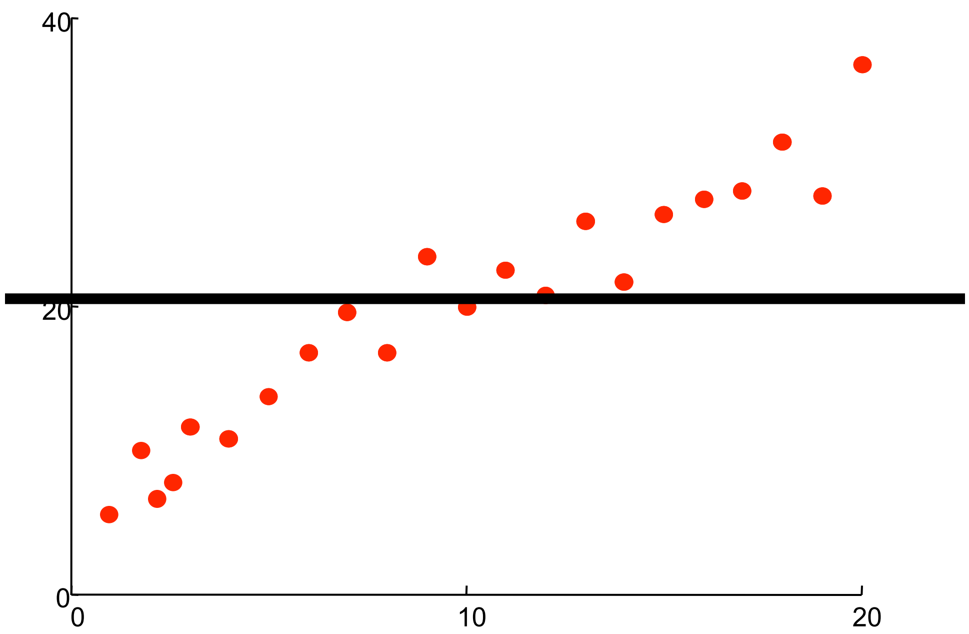


Measuring error

Misclassification rate: fraction of training data whose prediction is wrong
(Might also care about *type* of mistake…)



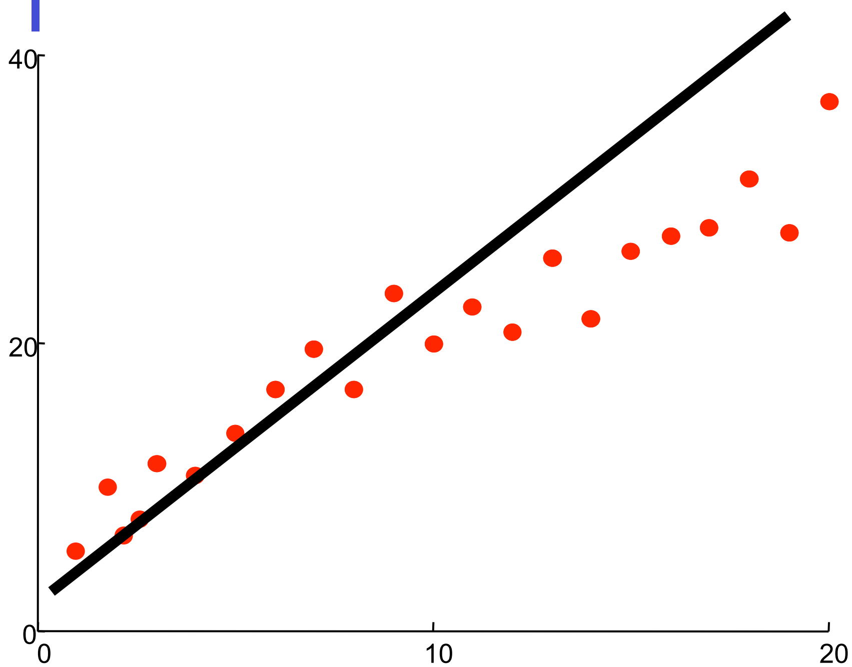
Measuring error



$$\hat{y}(x) = \theta_0 + \theta_1 x$$

- What makes a good predictor?

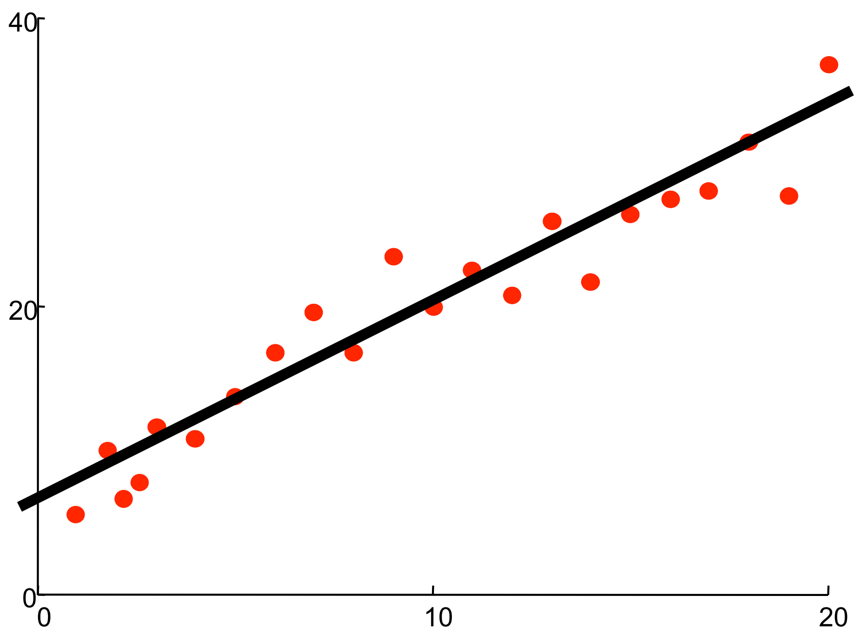
Measuring error



$$\hat{y}(x) = \theta_0 + \theta_1 x$$

- What makes a good predictor?

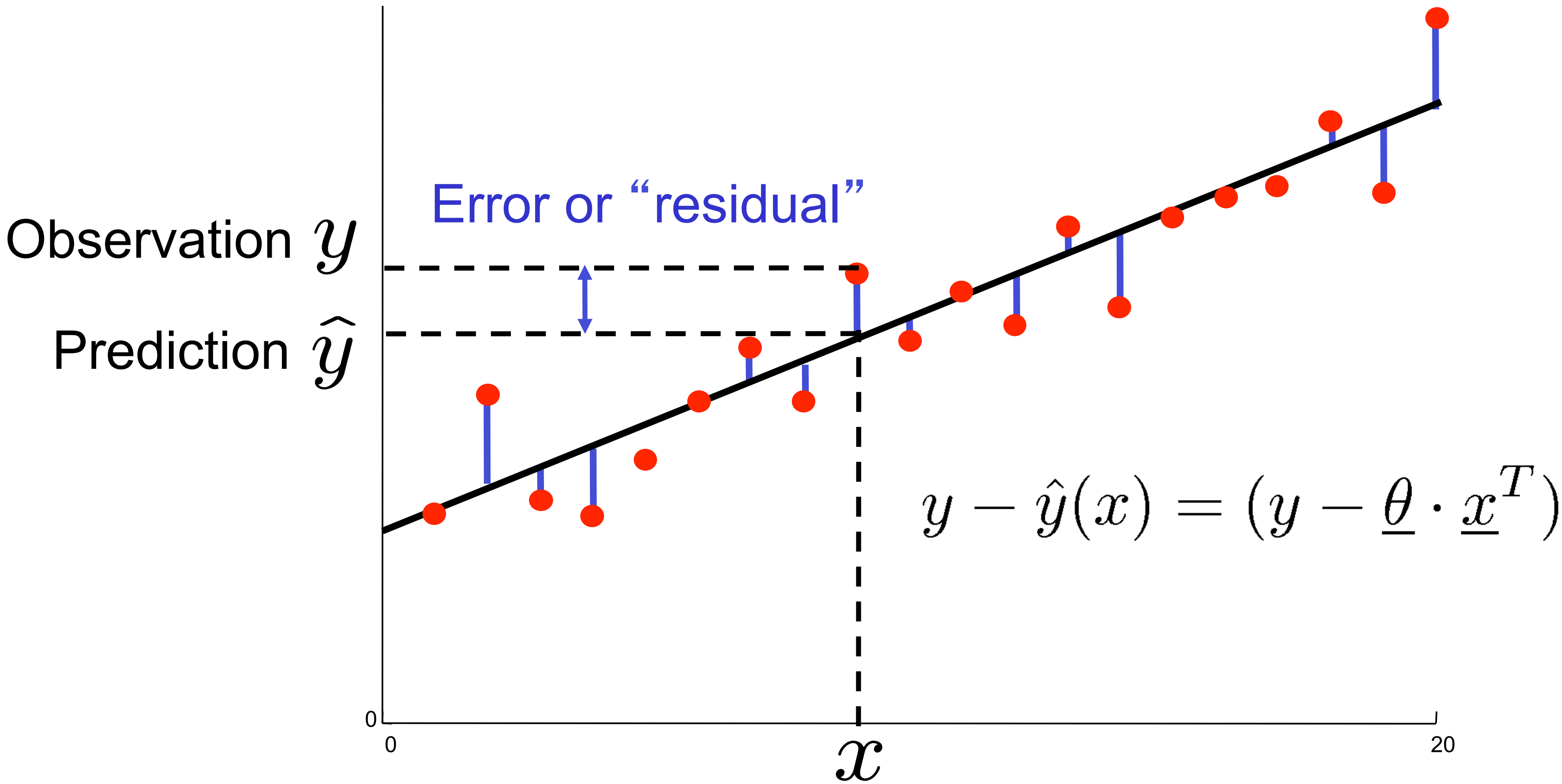
Measuring error



$$\hat{y}(x) = \theta_0 + \theta_1 x$$

- What makes a good predictor?

Measuring error



Sum of squared error

- How can we quantify the error?

$$\text{SSE},\ J(\underline{\theta}) = \frac{1}{2}\sum_j (y^{(j)} - \hat{y}(x^{(j)}))^2$$
$$= \frac{1}{2}\sum_j (y - \underline{\theta} \cdot \underline{x}^T)^2$$

- Could choose something else, of course…
  - Computationally convenient (more later)
  - Measures the variance of the residuals
  - Corresponds to Gaussian model of "noise"

$$\mathcal{N}(y\ ;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

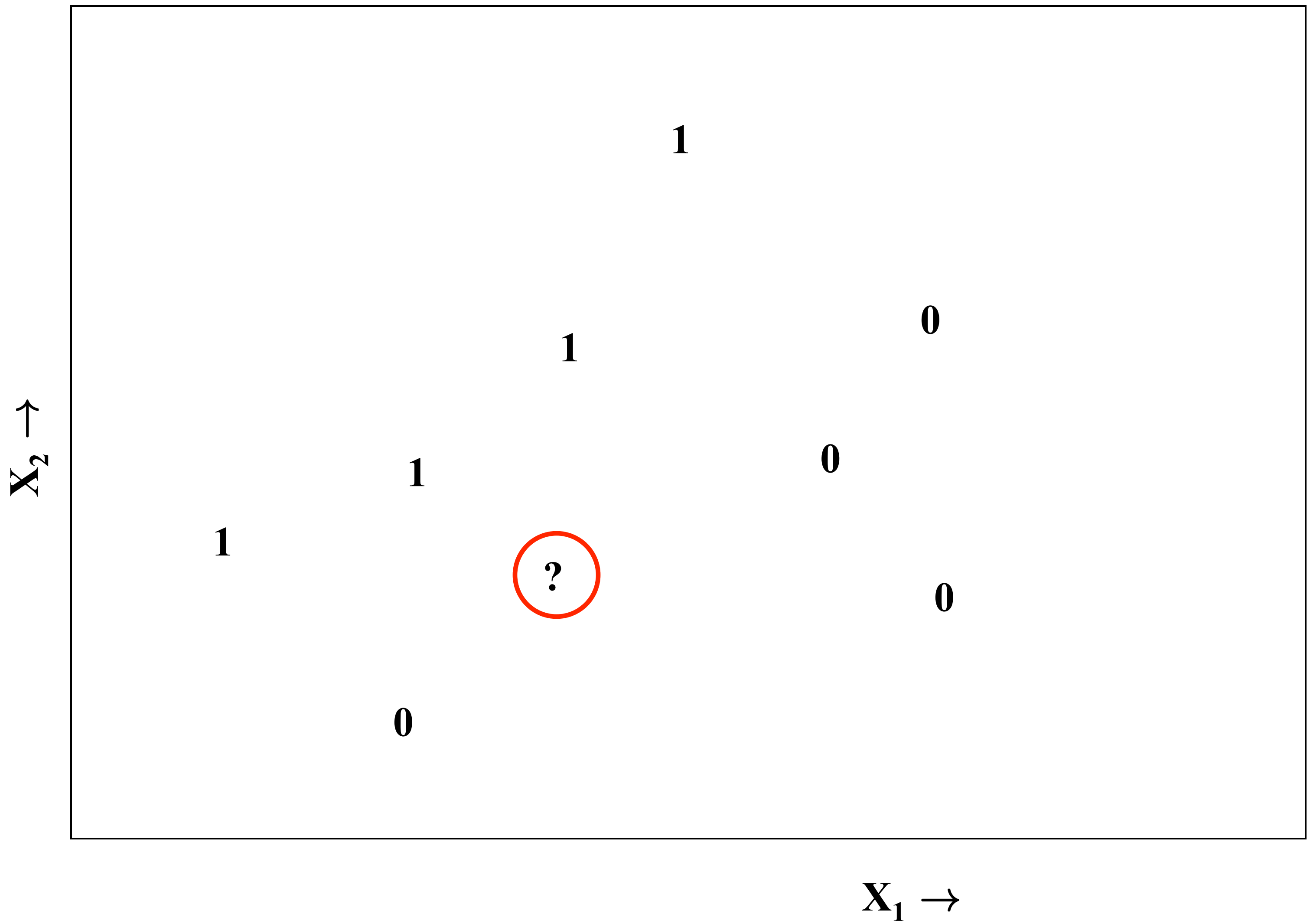
Nearest neighbor classifier

- $\underline{x}$ is a new feature vector whose class label is unknown
- Search training data for the closest feature vector to $\underline{x}$
  - Suppose the closest one is $\underline{x}^{(j)}$
- Classify $\underline{x}$ with the same label as $\underline{x}^{(j)}$, i.e.
  - Assign $\underline{x}$ the predicted label $\underline{y}^{(j)}$
- Interpretation as memorization
- How are "closest $\underline{x}$" vectors determined?
  - typically use minimum Euclidean distance

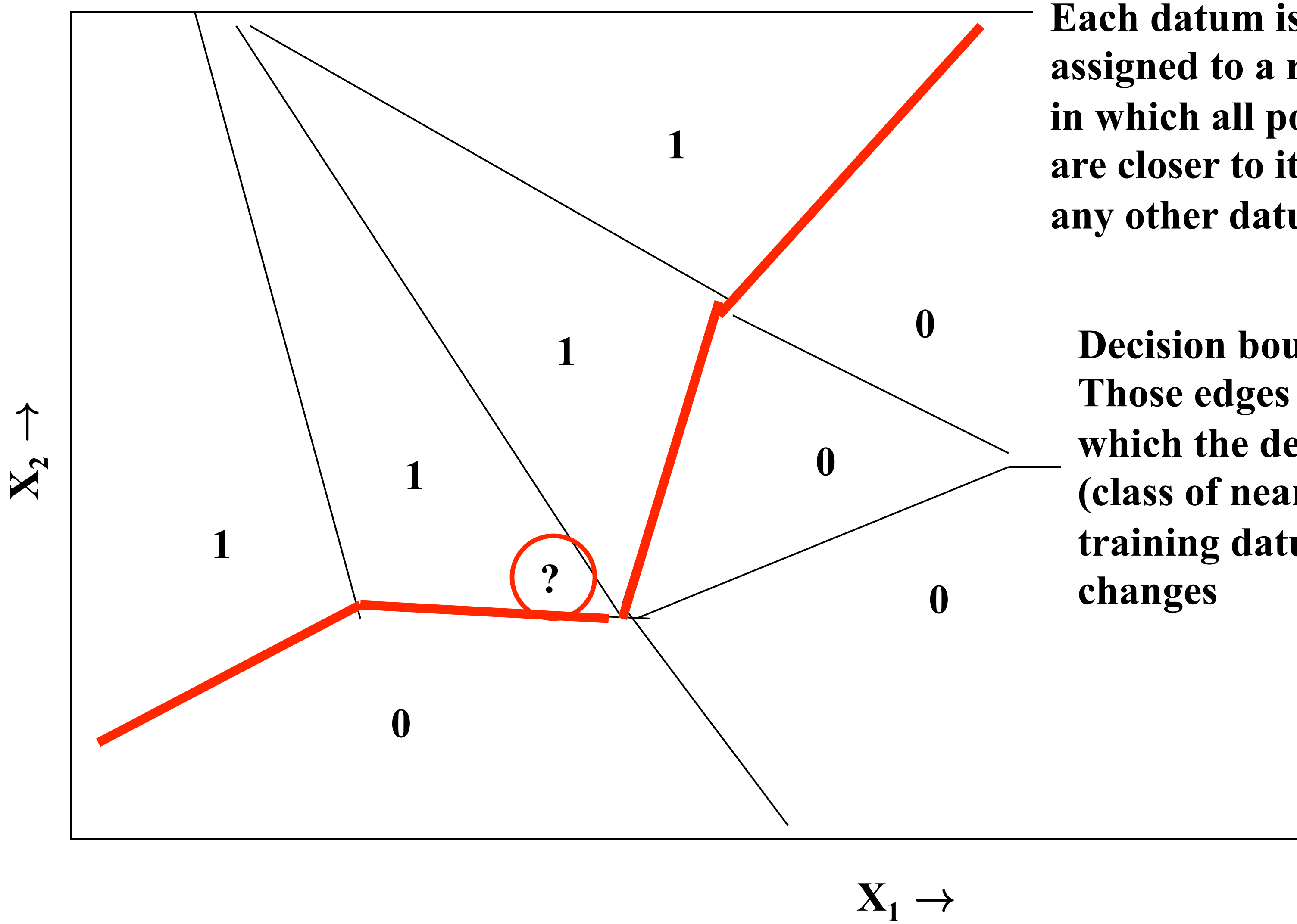
$$d(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$

- Side note: this produces a "Voronoi tesselation"
    - each point "claims" a cell surrounding it
    - cell boundaries are polygons

Nearest neighbor classifier



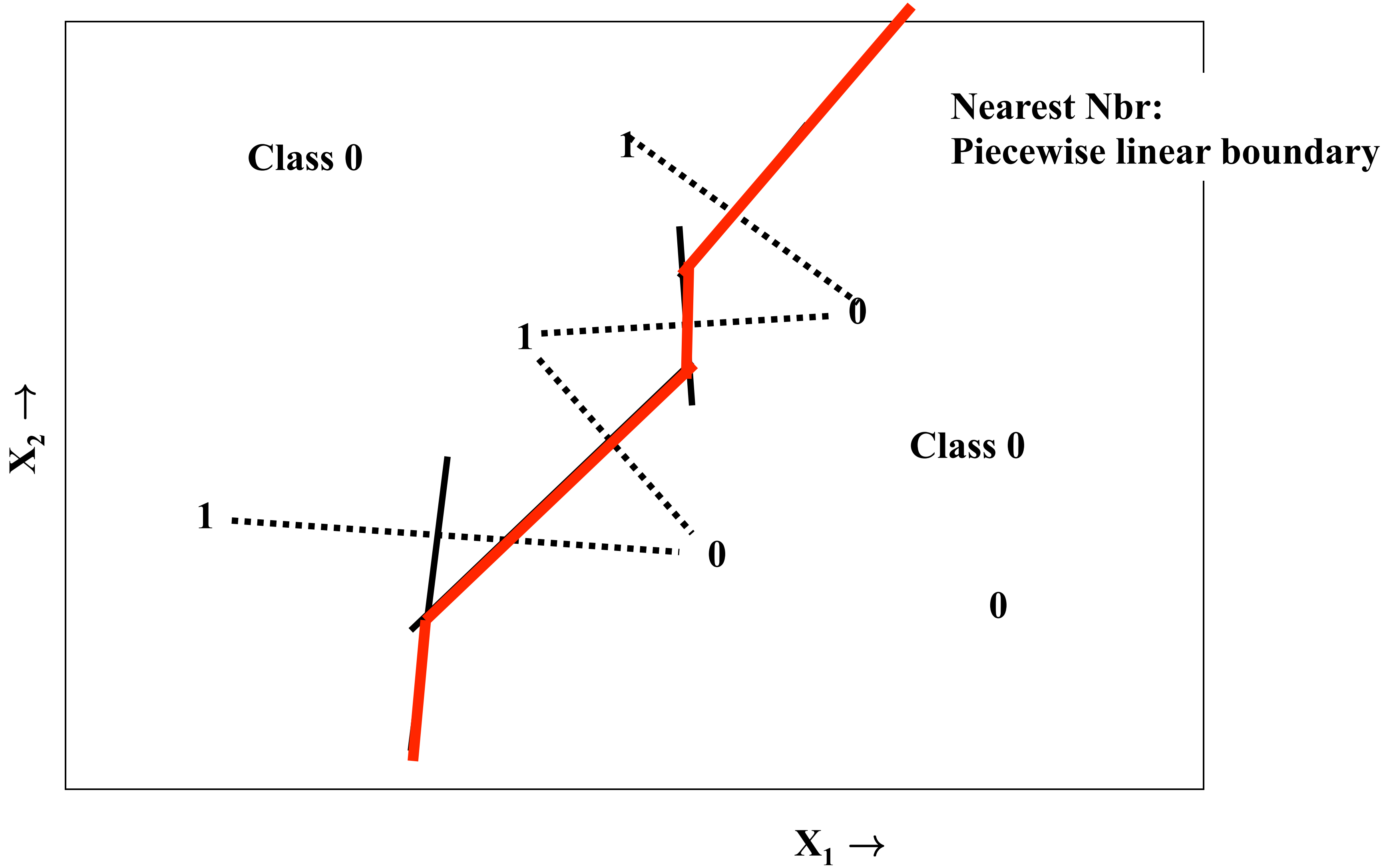
Nearest neighbor classifier



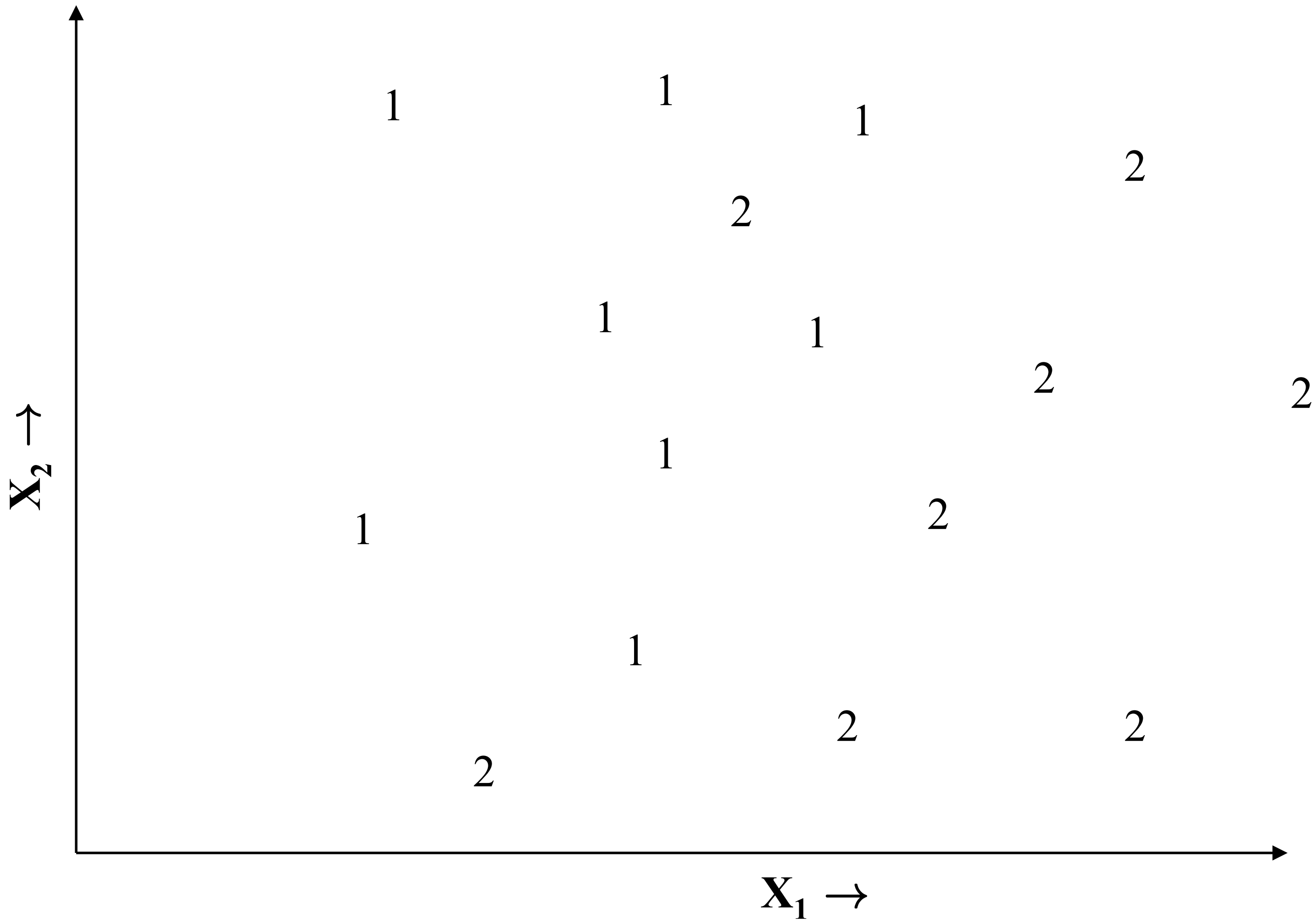
Each datum is assigned to a region, in which all points are closer to it than any other datum

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

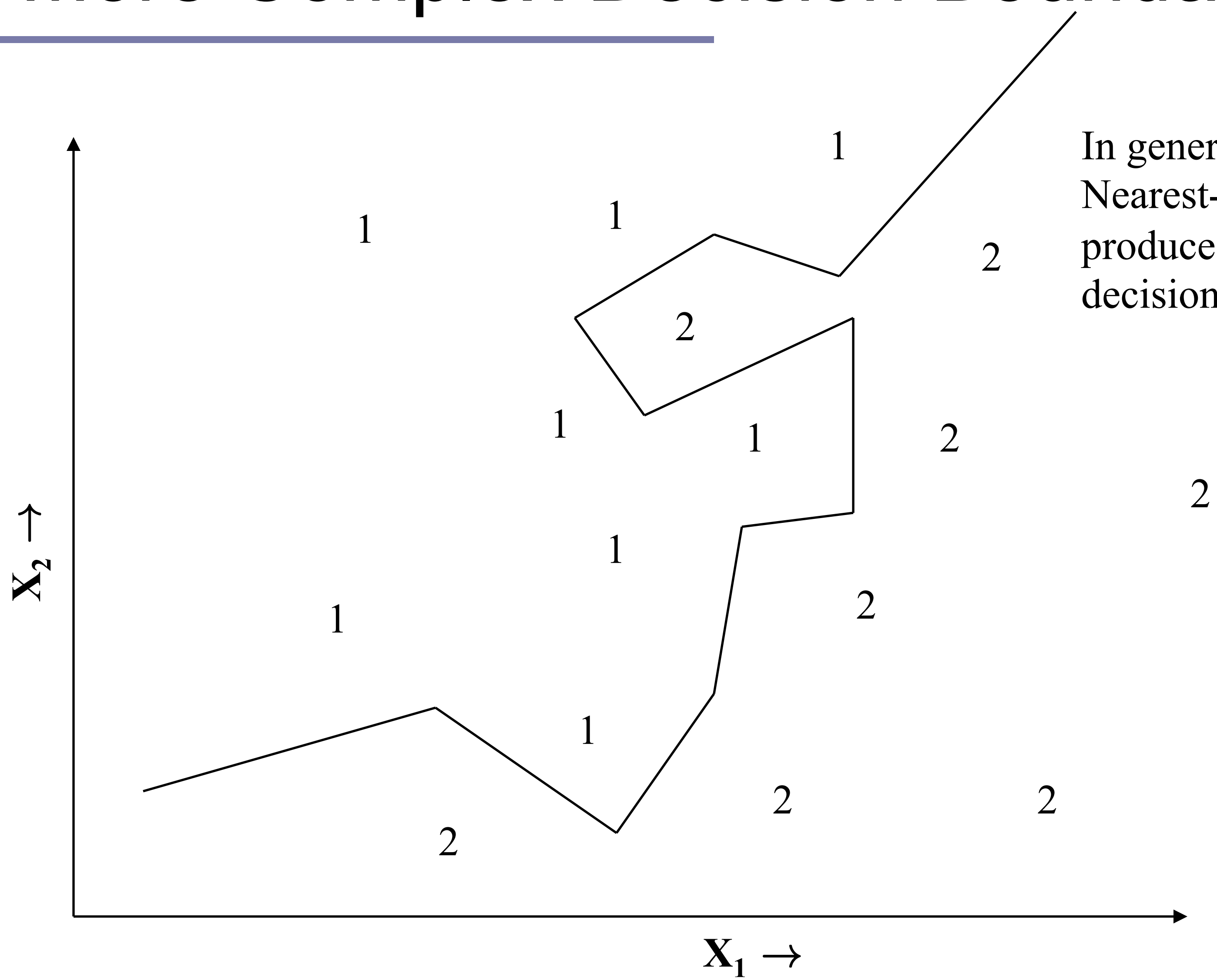
Nearest neighbor classifier



More Data Points



More Complex Decision Boundary



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

Summary

- What is ML; types of ML
  - Supervised Learning
- Definitions
- Cost functions
- K-nearest neighbor models
  - Classification (vote)
  - Regression (average or weighted avg)
- Piecewise linear decision boundary
  - How to calculate
- Test data and overfitting
  - Model "complexity" for knn
  - Validation data for test error rates