Introduction

Kevin Murphy Ch. 1

CS 6140
Machine Learning
Professor Olga Vitek

January 10, 2017

What is machine learning?

- Two general questions
  - How to construct computer systems that automatically improve through experience?
  - What are the fundamental statistical / computational / information-theoretical laws that govern all learning systems?

M. I Jordan and T. M. Mitchell. "Machine learning: Trends, perspectives, and prospects" *Science*, 349:255, 2015

Types of machine learning

- Supervised (predictive)
  - Data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$
  - **Goal:** find associations between $\mathbf{x}_i$ and y_i
  - $\mathbf{x}_i$: often D-dimensional vectors of attributes, features, covariates; in $N \times D$ design matrix
  - Can be arbitrary complex (image, sentence, graph)
  - y_i: response variable
  - When $y_i \in \{1, \ldots, C\}$ (categorical or nominal): classification or pattern recognition problem
  - When y_i continuous: regression

- Unsupervised (descriptive)
  - Data $\mathcal{D} = \{(\mathbf{x}_i)\}_{i=1}^N$
  - **Goal:** find "interesting patterns" in $\mathbf{x}_i$
  - $\mathbf{x}_i$ can also be arbitrary complex
  - Less well defined problem

Intermediate/blended types

- Reinforcement learning
  - An agent performs an action, and the environment returns a state (or reward)
  - The environment only gives an indication of whether the action was correct
  - Objective: maximize expected reward
    * E.g., robotics

- Semi-supervised learning
  - Uses unlabeled data to augment labeled data in a supervised learning context

More terms

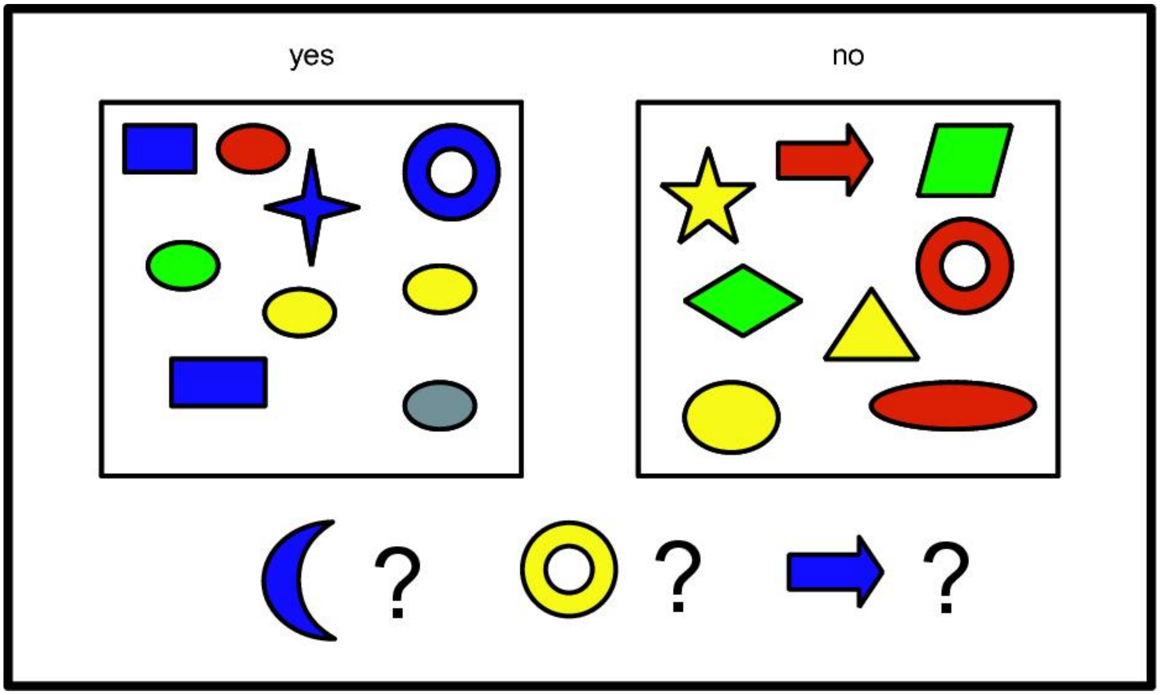
- Machine learning
  - Specific tasks associated with class discovery (unsupervised learning), class prediction (supervised learning)
- Data mining
  - Analysis of (often large) observational datasets to *find unexpected relationships*
  - Often secondary, exploratory analysis of convenience (opportunity) datasets
- Statistics
  - Collection and analysis of data, to *make inference beyond the current dataset*.
  - Characterized by *measures of uncertainty* , and of decision making in presence of uncertainty
  - Often primary, confirmatory analysis of designed experiments or ad-hoc datasets
- Data science
  - Often used interchangeably with data mining
  - Often used in 'data-driven decision making'

Supervised learning

Goals

- Training set:
  - Learn mapping from inputs $\mathbf{x}_i$ to outputs y_i
    * $y_i \in \{1, 2\}$: binary classification
    * $y_i \in \{1, \ldots, C\}, C > 2$: multiclass classification
- Formalize:
  - Approximate with unknown function $y = f(\mathbf{x})$
  - Make predictions $\widehat{y} = \widehat{f}(\mathbf{x})$
- Generalize:
  - Make predictions on novel inputs

Example: color shape classification



D features (attributes)

N cases

Color	Shape	Size (cm)
Blue	Square	10
Red	Ellipse	2.4
Red	Ellipse	20.7

Label
1
1
0

K. Murphy, Fig 1.1a

Example: color shape classification

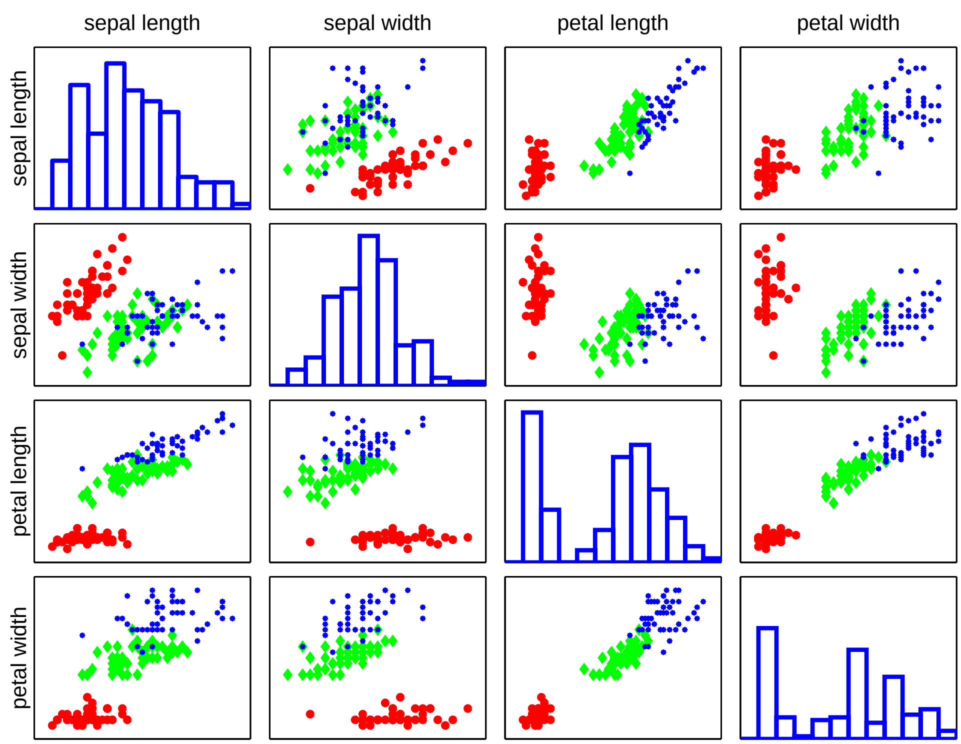
- Data
  - $y = \{0, 1\}$
  - $\mathbf{x}$: color, shape, size
- Generalize
  - Predict on cases are not part of the training set
  - Blue crescent 1? Yellow circle 0 or 1? Blue arrow?

Probabilistic predictions

- $p(y|\mathbf{x}, \mathcal{D})$: Probability distribution over y, conditional on $\mathbf{x}$ and $\mathcal{D}$
  - If y is binary, return $p(y = 1|\mathbf{x}, \mathcal{D})$
  - Implicitly conditions on the probability model that links $\mathbf{x}$ and y

- Maximum *a posteriori* (MAP) estimate
  - "Best guess" prediction: most probable label
  - Characterizes uncertainty in the prediction
  - $\hat{y} = \hat{f}(\mathbf{x}) = argmax_{c=1,\ldots,C}\, p(y = c|\mathbf{x}, \mathcal{D})$

Example: iris flower classification



K. Murphy, Fig 1.3 and 1.4

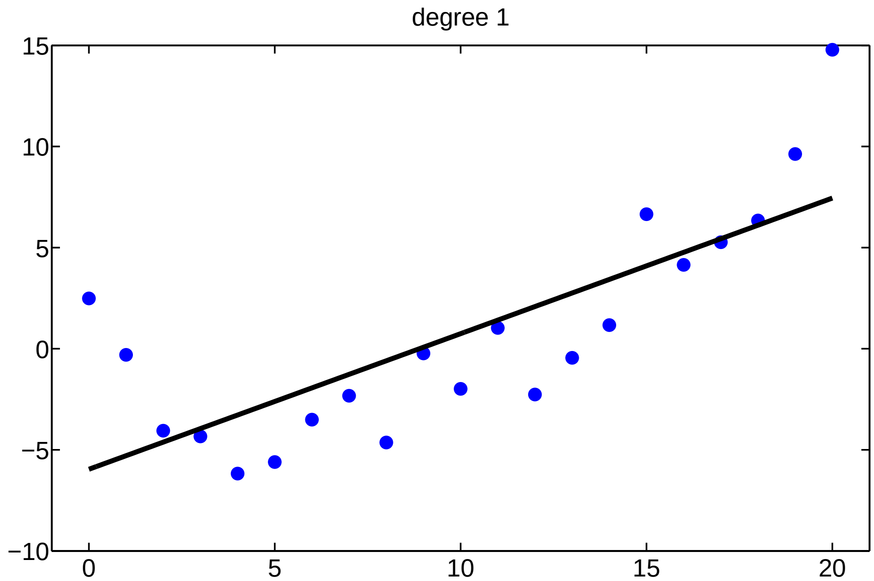
Examples

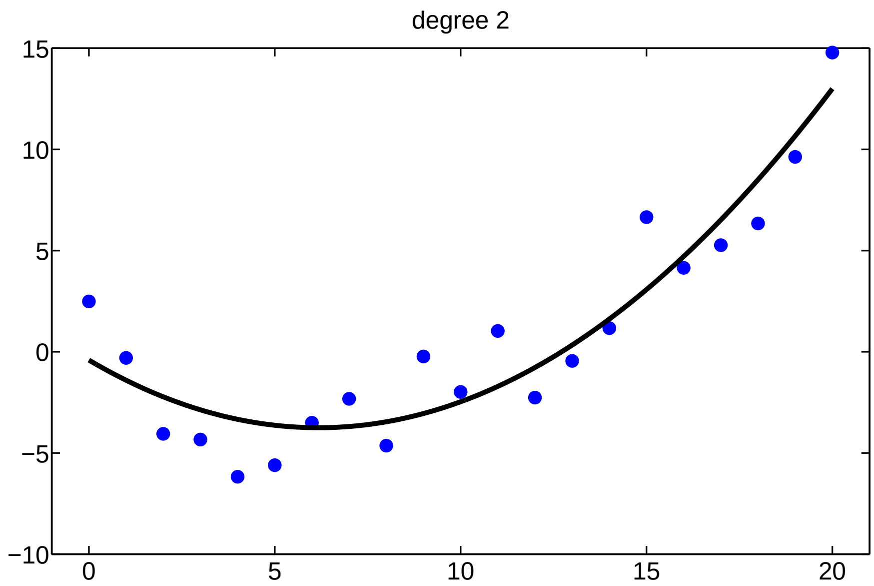
- Iris: challenges
  - Human feature extraction
  - Two species (blue and green circles) can only be distinguished by combination of two features
- Other examples
  - Document classification (e.g., spam detection)
  - Fraud detection
  - Image, object, speech classification
  - Early detection of disease, and prediction of therapy response
  - ....

Regression

- Training set:
  - Learn mapping from inputs $\mathbf{x}_i$ to outputs y_i
    - $\mathbf{x}_i$ continuous
- Examples
  - Predict age of YouTube viewer
  - Predict temperature inside a building using weather and sensor data
  - Predict amount of prostate specific antigen (PSA) in the body from clinical measurements

Example





K. Murphy, Fig 1.7

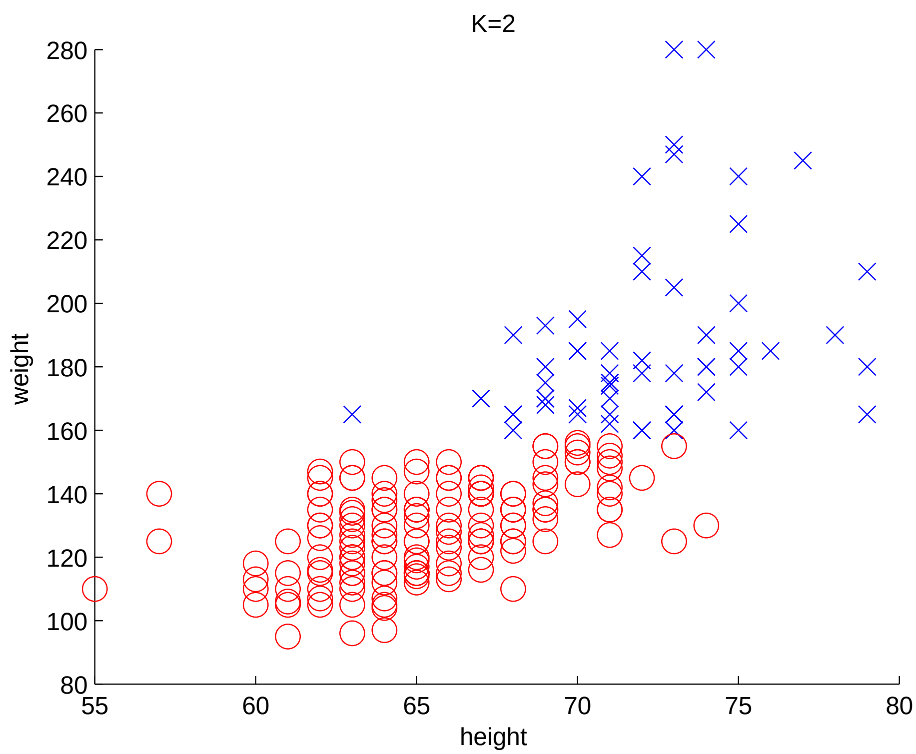
Unsupervised learning
(knowledge discovery)

Goals

- Training set:
  - Learn patterns in $\mathbf{x}_i$
- Formalize:
  - Unconditional density estimation of form $p(\mathbf{x}_i|\theta)$
  - Multivariate density in the feature space
- Examples of tasks
  - Discovering clusters
  - Discovering latent factors
  - Discovering graph structure
    - Find pairs of highly correlated items: correlated stocks, correlated patterns of behavior...
  - Missing value imputation
  - Image implainting

Example

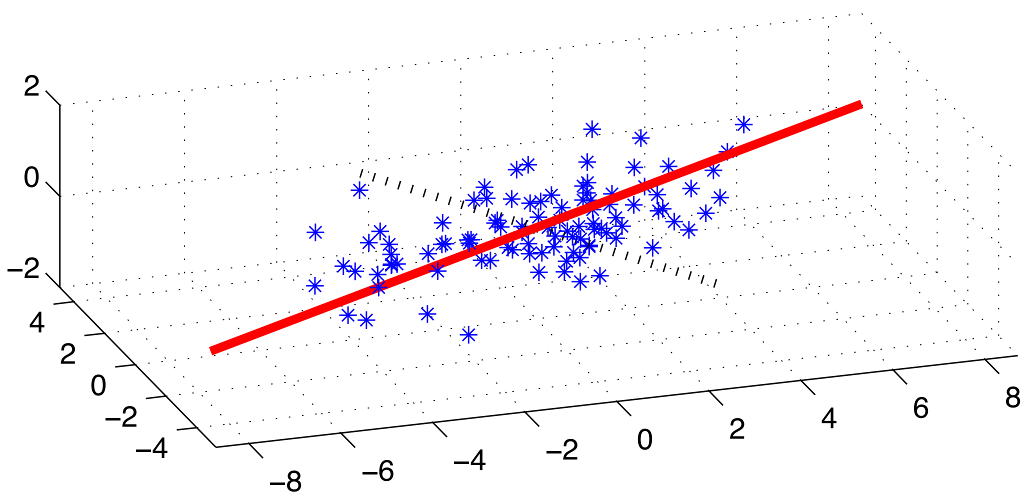
Discovering clusters of people by weight and height

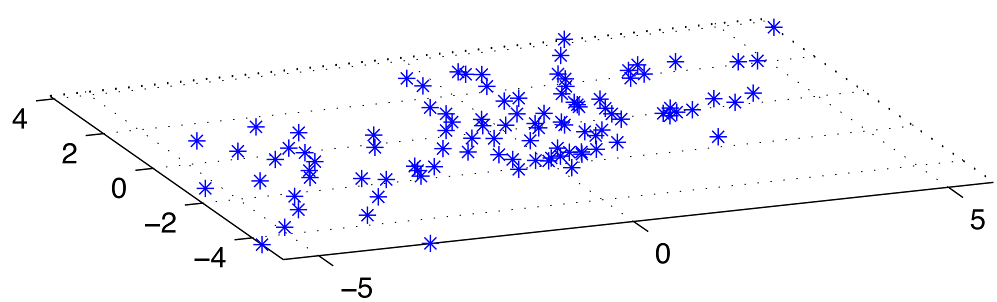


K. Murphy, Fig 1.8

Example

Discovering latent factors: dimensionality reduction





K. Murphy, Fig 1.9

Basic concepts in machine learning

Parametric vs non-parametric models

Parametric vs non-parametric models

- How to specify $p(y|\mathbf{x})$ or $p(\mathbf{x})$

- Parametric:

  * Functions of a fixed (often relatively small) number of parameters

  * **Example: linear regression**

  + Simpler interpretation, faster, better performance if correct assumptions

  − Restrictive assumptions

- Non-parametric:

  * Large # of parameters, often grows with with size of training data

  * **Example: K-nearest neighbor (KNN)**

  + Flexibility

  − Not assumption-free, slower, loss of accuracy if not making relevant assumptions

Linear regression

- ML: $y(\mathbf{x}) = \sum_{j=1}^{D} w_j x_j + \epsilon$
  - $\mathbf{w}$: weights

- Statistics: $y(\mathbf{x}) = \sum_{j=1}^{D} \beta_j x_j + \epsilon$
  - β: parameters

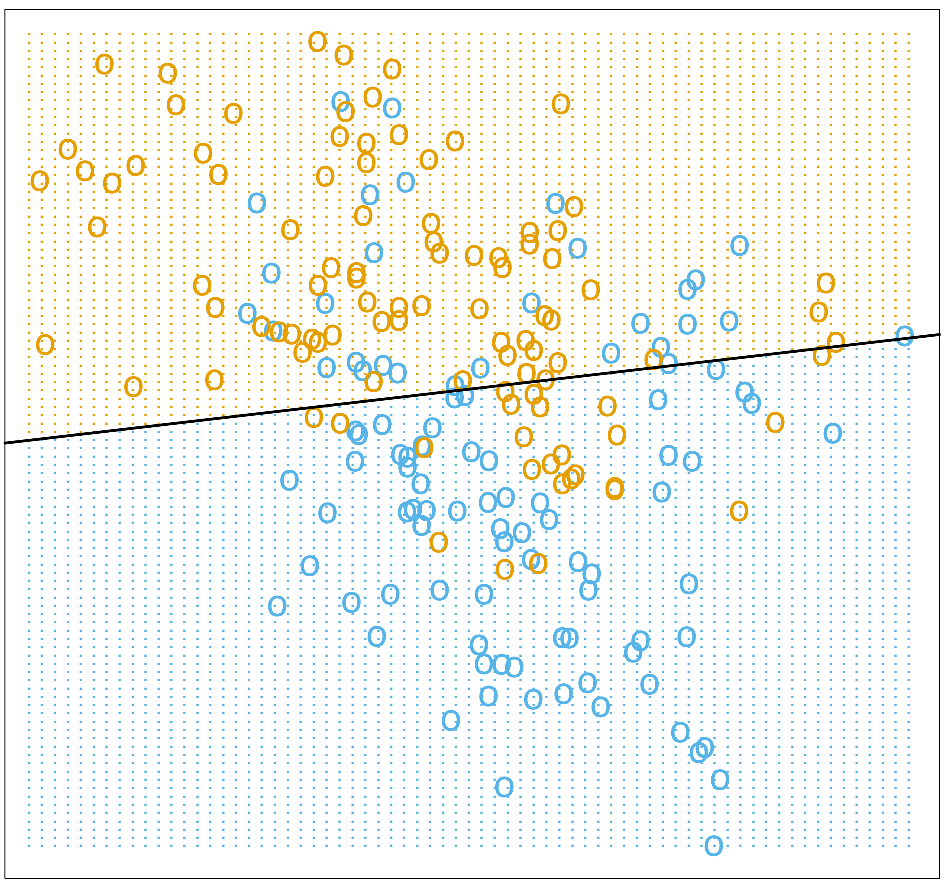
- Parameter estimation: least squares
  - $\widehat{\beta} = argmin_{\beta} \sum_{i=1}^{N} (y_i - x_i'\beta)^2$

- Probabilistic prediction:
  - ϵ: residual error, $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$
  - $\theta = (\beta, \sigma^2)$
  - $p(y|\mathbf{x}, \theta) = \mathcal{N}(x_i'\beta, \sigma^2)$

Linear regression

$$y = \begin{cases} 0, & \text{if blue} \\ 1, & \text{if orange} \end{cases} \quad \hat{y} = \begin{cases} \text{blue}, & \text{if } \hat{y} \le 0.5 \\ \text{orange}, & \text{if } \hat{y} \ge 0.5 \end{cases}$$

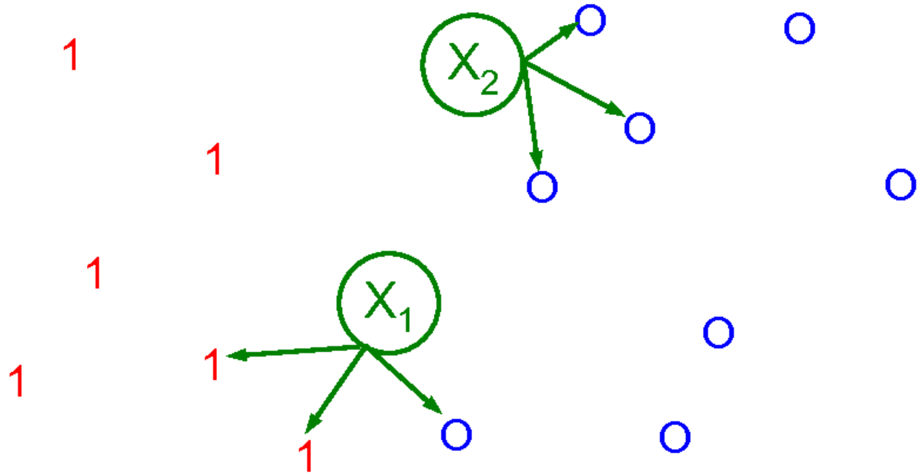
Black line: decision boundary, relies on linearity (here likely suboptimal)



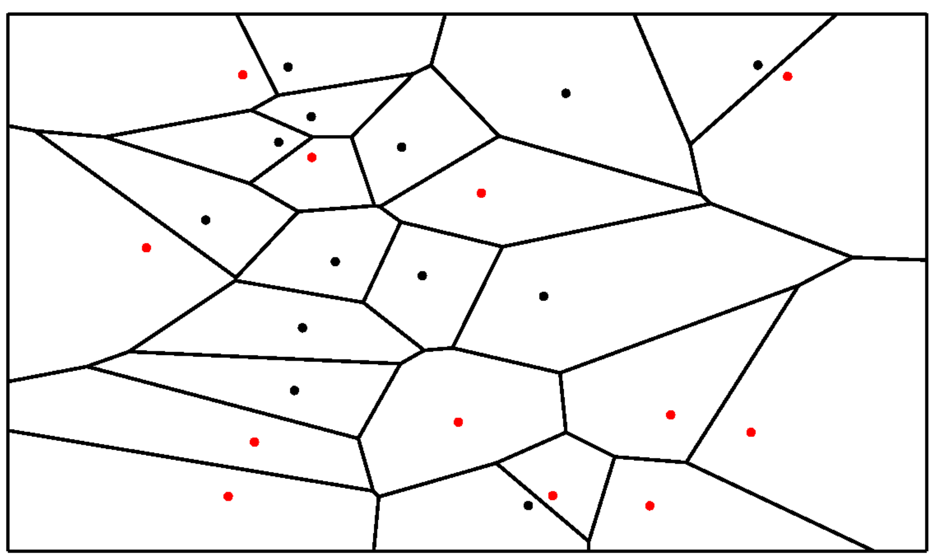
Hastie, Tibshirani, Friedman, Fig 2.1

KNN

K nearest neighbors, 2 predictors, $K = 3$



Voronoi tesselation, $K = 1$



K. Murphy, Fig 1.14

KNN

- Example of memory-based learning or instance-based learning

- $p(y = c|\mathbf{x}, \mathcal{D}, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x},\mathcal{D})} I(y_i = c)$

  - Assigns label to $y(\mathbf{x})$ according to the majority of K nearest labels in training set

  - $N_K(\mathbf{x}, \mathcal{D})$ is the K-neighborhood

  - Closeness requires a metric (e.g., Eucledian distance)

  - I is the indicator function

- Apparent parameter: K
- Effective number of parameters: N/K

  - Number of non-overlapping neighborhoods with same mean

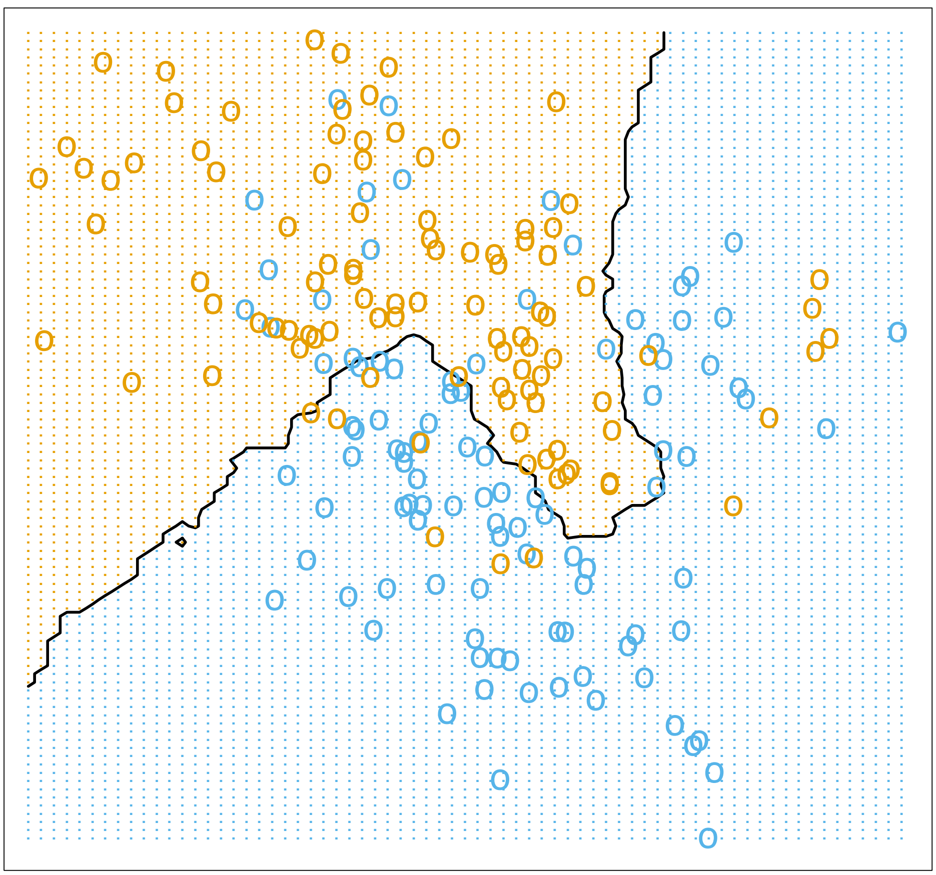
  - Grows with N for a fixed K

  - Least squares not appropriate

KNN

K nearest neighbors, 2 predictors, $K = 15$

Fewer misclassifications than linear regression

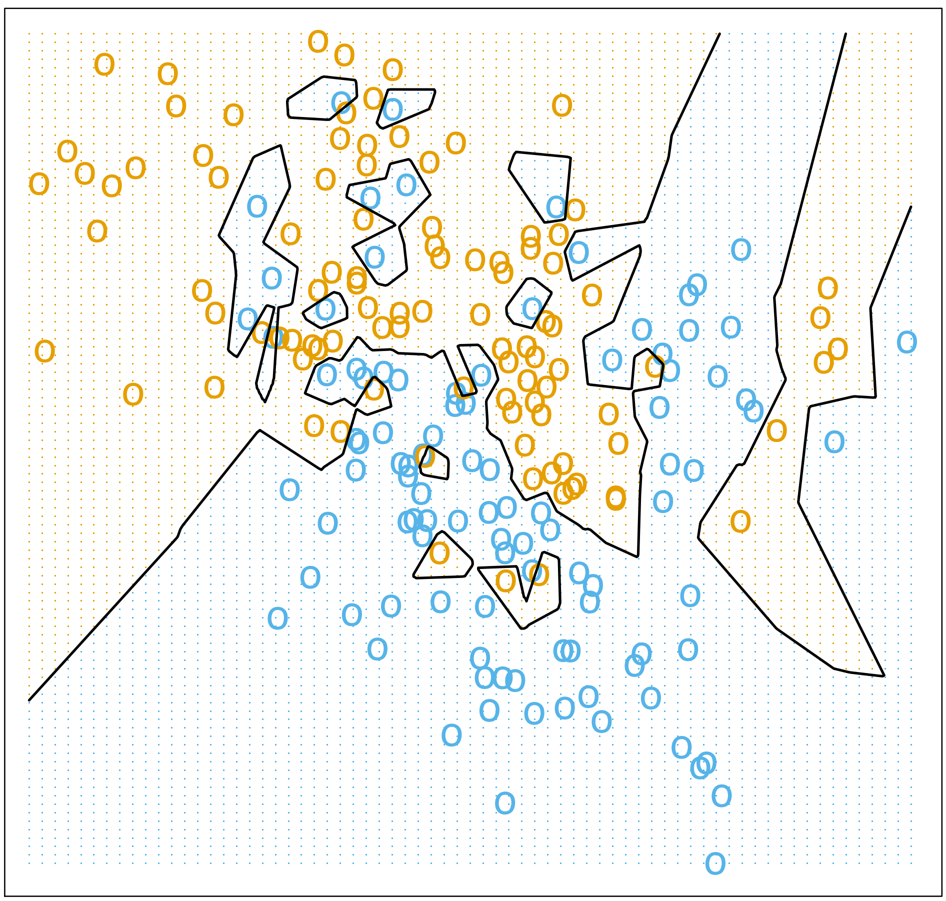


Hastie, Tibshirani, Friedman, Fig 2.2

KNN

K nearest neighbors, 2 predictors, $K = 1$

No misclassifications, flexible



Hastie, Tibshirani, Friedman, Fig 2.3

Overfitting and model selection

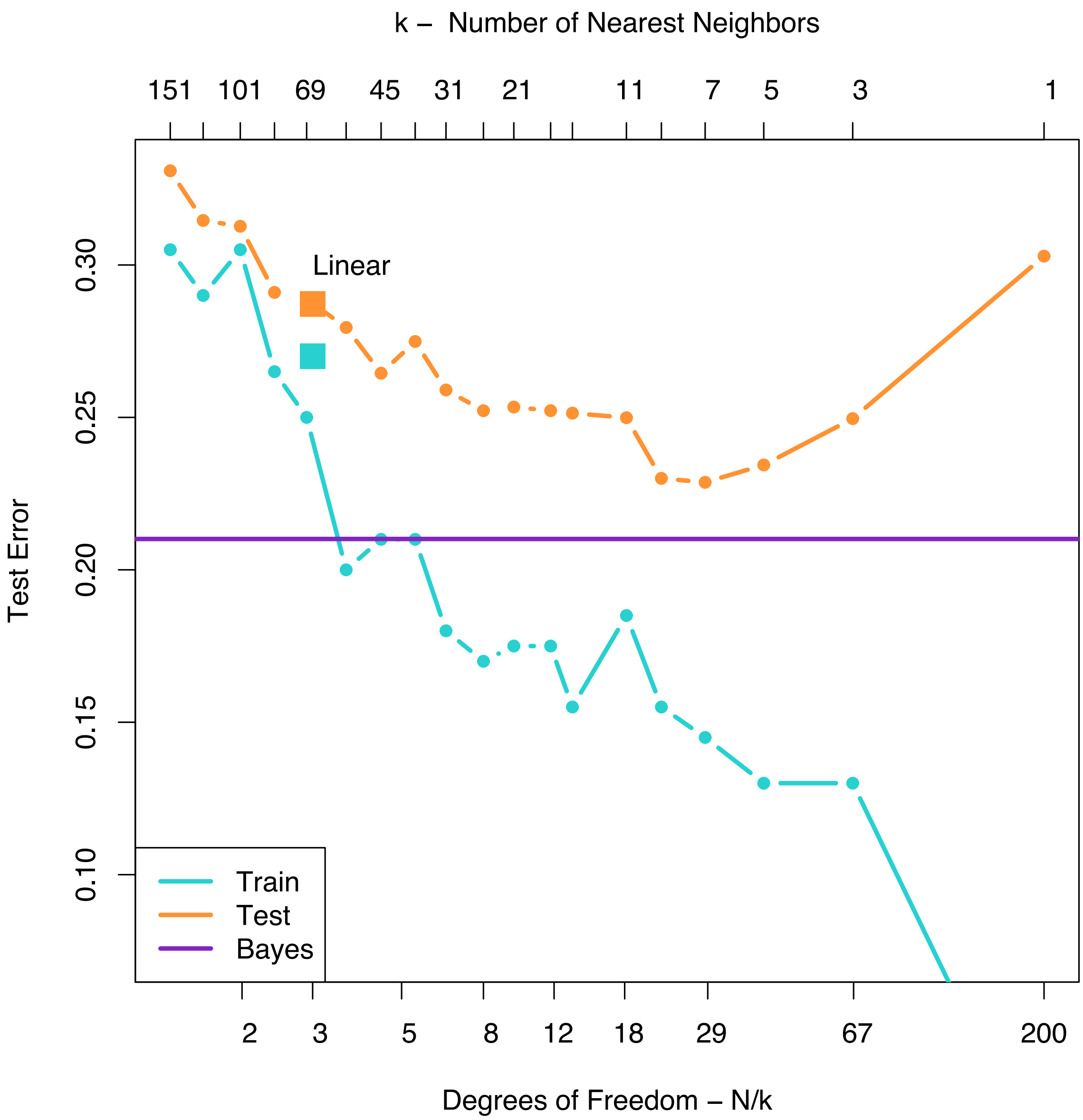
- Model representation $f(\mathbf{x})$ fits too closely to the training set
  - Typically: too many parameters

- Define misclassification rate
  - $err(f, D) = \frac{1}{N} \sum_{i=1}^{N} I(f(\mathbf{x}_i) \neq y_i)$
  - $\uparrow$ # parameters $\Rightarrow \downarrow err(f, D)$ on training set

- Define generalization error rate
  - Average misclassification rate over future data
  - Use independent test set or cross-validation
  - Pick # parameters minimizing generalization error rate

Evaluation

Predictive performance on 10,000 independent validation observations



Hastie, Tibshirani, Friedman, Fig 2.4

Local methods in high dimensions

Curse of dimensionality

Curse of dimensionality

- Methods such as KNN do not work well with high-dimensional inputs

- Example:

  - Uniformly distribute inputs in D-dim. unit cube

  - Take a target data point $\mathbf{x}$

  - Build a small hypercube around it until it captures a fixed fraction r of data points

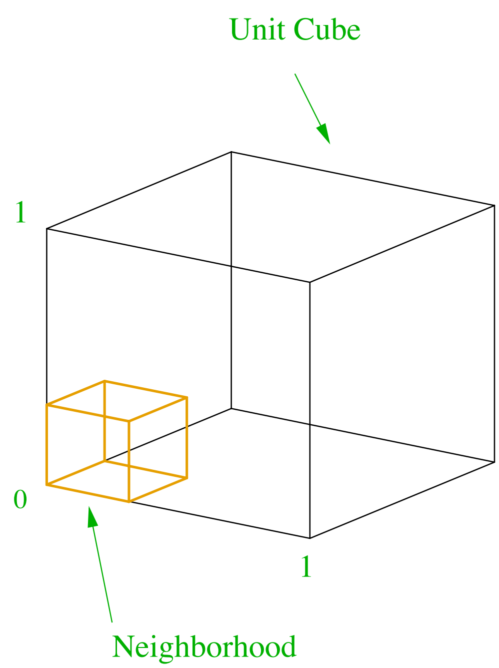
  - This corresponds to fraction r of unit volume

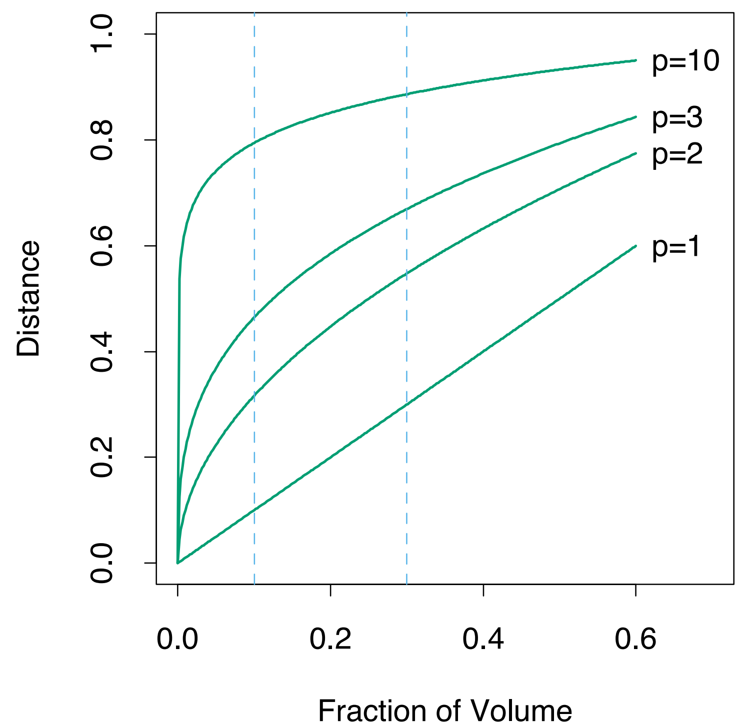
  - Expected length of the edge of the small hypercube is $e_D(r) = r^{1/D}$

  - If $D = 10$, and neighborhood is to contain 10% of data, extend the small hypercube by 80% in each direction

    $\Rightarrow$ Points so far away in each dimension may not predict well the label of $\mathbf{x}$

Curse of dimensionality





Hastie, Tibshirani, Friedman, Fig 2.6

Curse of dimensionality

- All sample points are close to the edge of the sample

- Example:

  – Uniformly distribute inputs in D-dim. unit ball, centered at origin

  – Consider a nearest neighbor at origin

  – The median distance from the origin to the closest data point is

$$d(D, N) = \left(1 - \frac{1}{2}^{1/N}\right)^{1/D}$$

  – If $N = 500$ and $D = 10$, the median distance $d(D, N) \approx 0.52$

    $\Rightarrow$ Most data points are closer to the boundary than to the origin

    $\Rightarrow$ Prediction must extrapolate from neighboring sample, rather than interpolate

No free lunch theorem

- No single best model works optimally for all problems
  - (Wolpert 1996)

- Different models are needed for different real-world problems

Challenges and opportunities

- New problem formulations
  - Specialized domain knowledge
  - New sources of data
  - Data size and computer architecture
- Resource constraints
  - Privacy
  - Communication (e.g., data aggregation)
- Mimic human behavior
  - Multiple tasks
  - Continuous simple-to-difficult learning
  - Team based / mixed-initiative learning