



Cornell University

# Instance-Based Learning

CS4780/5780 – Machine Learning  
Fall 2011

Thorsten Joachims  
Cornell University

Reading: Mitchell Chapter 1 & Sections 8.1 - 8.2



- **Definition:**  
Acquire an operational definition of a general category of objects given positive and negative training examples.



<b>correct</b> <b>(3)</b>	<b>color</b> <b>(2)</b>	<b>original</b> <b>(2)</b>	<b>presentation</b> <b>(3)</b>	<b>binder</b> <b>(2)</b>	<b>A+Homework</b>
complete	yes	yes	clear	no	yes
complete	no	yes	clear	no	yes
partial	yes	no	unclear	no	no
complete	yes	yes	clear	yes	yes

**Instance Space X:** Set of all possible objects described by attributes (often called features).

**Concept c:** Subset of objects from X (c is unknown).

**Target Function f:** Characteristic function indicating membership in c based on attributes (i.e. label)(f is unknown).

**Training Data S:** Set of instances labeled with target function.



# Concept Learning as Learning a Binary Function

- Task:
  - Learn (to imitate) a function  $f: X \rightarrow \{+1, -1\}$
- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs  $\rightarrow$  **training examples**
  - An **example** is a pair  $(x, y)$ , where  $x$  is the input and  $y=f(x)$  is the output of the target function applied to  $x$ .
- Goal:
  - Find a function
$$h: X \rightarrow \{+1, -1\}$$
that approximates
$$f: X \rightarrow \{+1, -1\}$$
as well as possible.



- Given: Training data  $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ 
  - Attribute vectors:  $\vec{x}_i \in X$
  - Label:  $y_i \in Y = \{-1, +1\}$
- Parameter:
  - Similarity function:  $K : X \times X \longrightarrow \mathfrak{R}$
  - Number of nearest neighbors to consider:  $k$
- Prediction rule
  - New example  $x'$  with
  - K-nearest neighbors:  $k$  train examples with largest  $K(\vec{x}_i, \vec{x}')$

$$h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} 1_{[y_i=y]} \right\}$$



	<b>correct (3)</b>	<b>color (2)</b>	<b>original (2)</b>	<b>presentation (3)</b>	<b>binder (2)</b>	<b>A+Homework</b>
1	complete	yes	yes	clear	no	yes / +1
2	complete	no	yes	clear	no	yes / +1
3	partial	yes	no	unclear	no	no / -1
4	complete	yes	yes	clear	yes	yes / +1

- How will new examples be classified?
  - Similarity function?
  - Value of  $k$ ?

$$h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} 1_{[y_i=y]} \right\}$$



- Given: Training data  $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ 
  - Attribute vectors:  $\vec{x}_i \in X$
  - Target attribute:  $y_i \in \{-1, +1\}$
- Parameter:
  - Similarity function:  $K : X \times X \longrightarrow \mathbb{R}$
  - Number of nearest neighbors to consider:  $k$
- Prediction rule
  - New example  $x'$
  - K-nearest neighbors:  $k$  train examples with largest  $K(\vec{x}_i, \vec{x}')$

$$h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} \mathbf{1}_{[y_i=y]} K(\vec{x}_i, \vec{x}') \right\}$$

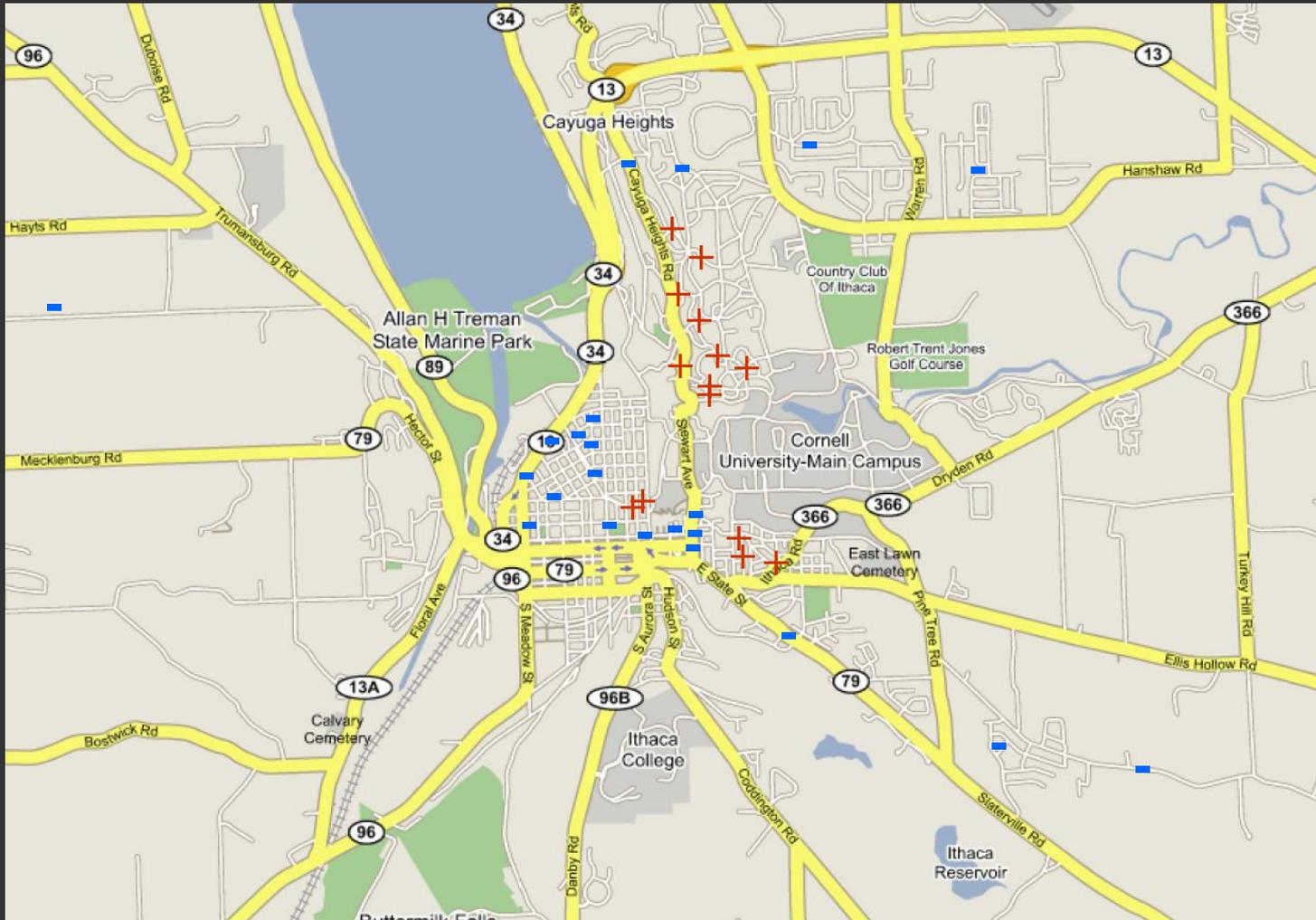


- Symbolic (nominal)
  - *EyeColor* {*brown, blue, green*}
- Boolean
  - *alive* {*TRUE, FALSE*}
- Numeric
  - Integer: *age* [0, 105]
  - Real: *length*
- Structural
  - Natural language sentence: parse tree
  - Protein: sequence of amino acids



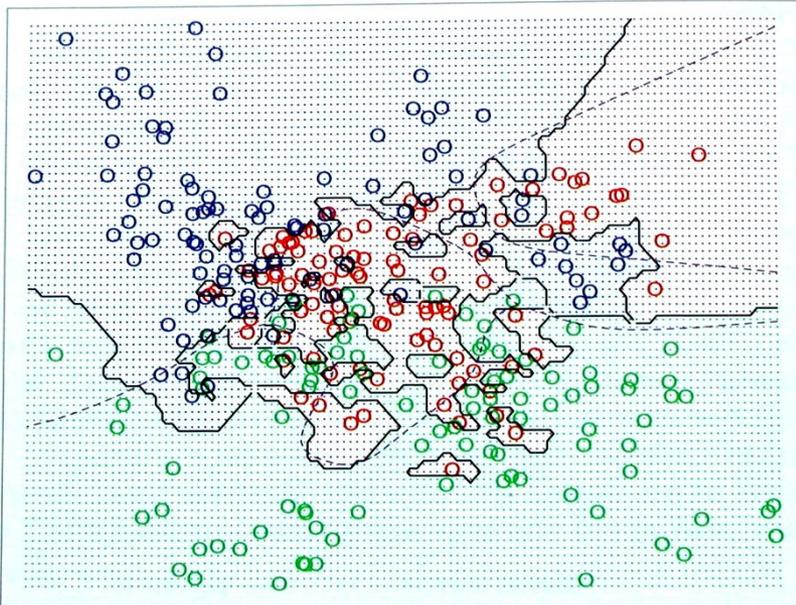
Cornell University

# Example: Expensive Housing (>\$200 / sqft)

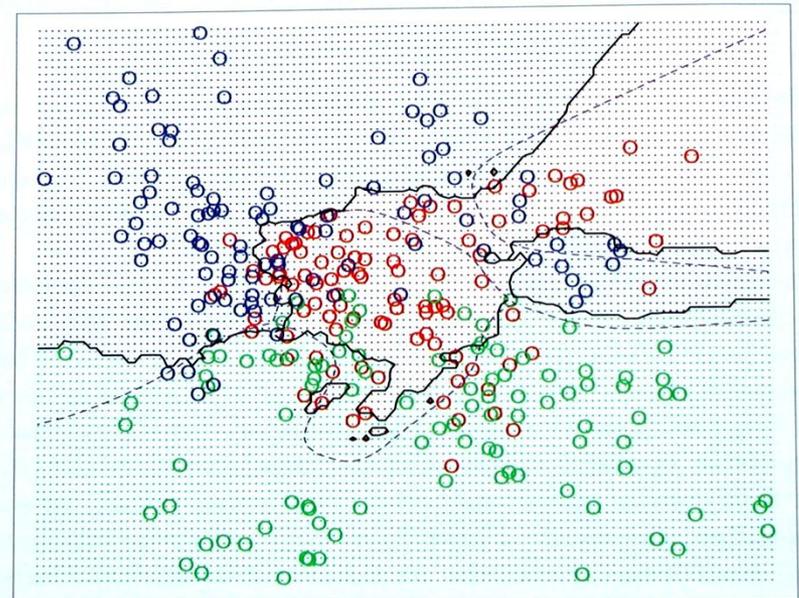




1-Nearest Neighbor



15-Nearest Neighbors





- Task:
  - Learn (to imitate) a function  $f: X \rightarrow Y$
- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs  $\rightarrow$  **training examples**
  - An **example** is a pair  $(x, f(x))$ , where  $x$  is the input and  $f(x)$  is the output of the function applied to  $x$ .
- Goal:
  - Find a function
$$h: X \rightarrow Y$$
that approximates
$$f: X \rightarrow Y$$
as well as possible.



# Weighted K-Nearest Neighbor for Regression

- Given: Training data  $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ 
  - Attribute vectors:  $\vec{x}_i \in X$
  - Target attribute:  $y_i \in \mathcal{R}$
- Parameter:
  - Similarity function:  $K : X \times X \longrightarrow \mathcal{R}$
  - Number of nearest neighbors to consider:  $k$
- Prediction rule
  - New example  $x'$
  - K-nearest neighbors:  $k$  train examples with largest  $K(\vec{x}_i, \vec{x}')$

$$h(\vec{x}') = \frac{\sum_{i \in k\text{nn}(\vec{x}')} y_i K(\vec{x}_i, \vec{x}')}{\sum_{i \in k\text{nn}(\vec{x}')} K(\vec{x}_i, \vec{x}')}$$



The screenshot shows the Netflix website interface. At the top, there's a navigation bar with the Netflix logo and user information: "Thorsten Joachims | Your Account & Help". Below this is a menu with "Watch Instantly", "Just for Kids", "Browse DVDs", "Your Queue", and "★ Suggestions For You". A search bar is on the right. The main content area features a recommendation section titled "Based on your rating, we think you'll enjoy these titles". It includes a poll: "Want more suggestions? How often do you watch?" with options "Never", "Sometimes", and "Often" for categories "Goofy" and "Raunchy". Three movie posters are shown: "Pulling", "high life", and "LEAVES OF GRASS", each with a 5-star rating. Below this are two more sections: "Recently Watched" with "TRAILER PARK BOYS" and "Top 10 for Thorsten" with "THE LAST ENEMY", "GEORGE GENTLY", "MI-5", and "LOVE THE BEAST!".