

Instance-Based Learning

CS4780/5780 – Machine Learning
Fall 2012

Thorsten Joachims
Cornell University

Reading: Mitchell Chapter 1 & Sections 8.1 - 8.2

Concept Learning

- **Definition:**

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

Concept Learning Example

correct (3)	color (2)	original (2)	presentation (3)	binder (2)	A+
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 describable 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.

K-Nearest Neighbor (KNN)

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

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

KNN Example

	correct (3)	color (2)	original (2)	presentation (3)	binder (2)	A+
1	complete	yes	yes	clear	no	yes
2	complete	no	yes	clear	no	yes
3	partial	yes	no	unclear	no	no
4	complete	yes	yes	clear	yes	yes

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

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

Weighted K-Nearest Neighbor

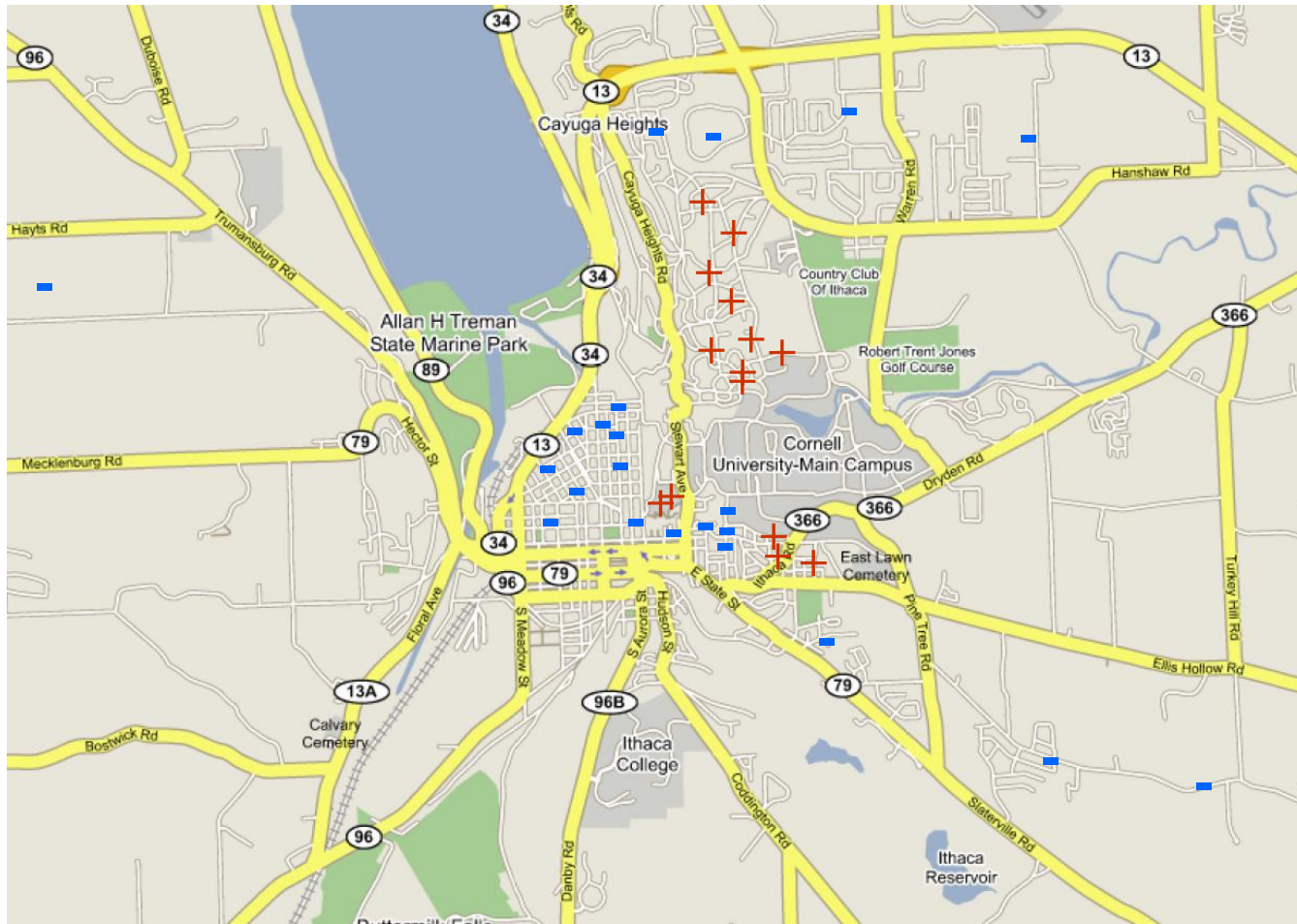
- Given: Training data $((\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n))$
 - Attribute vectors: $\vec{x}_i \in X$
 - Target attribute: $y_i \in Y$
- Parameter:
 - Similarity function: $K : X \times X \rightarrow \mathfrak{R}$
 - Number of nearest neighbors to consider: k
- Prediction rule
 - New example x'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

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

Types of Attributes

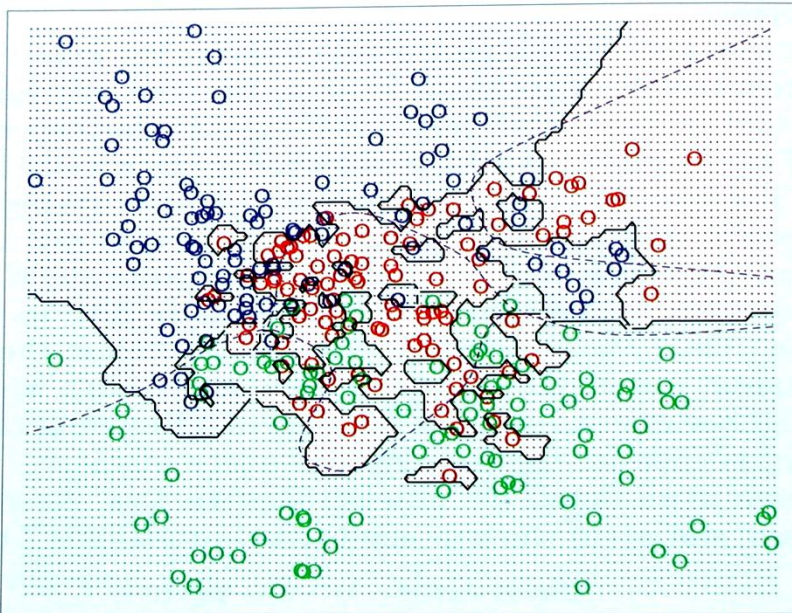
- Symbolic (nominal)
 - *EyeColor* {*brown, blue, green*}
- Boolean
 - *alive* {*TRUE, FALSE*}
- Numeric
 - Integer: *age* [0, 105]
 - Real: *length*
- Structural
 - Natural language sentence: parse tree
 - Protein: sequence of amino acids

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

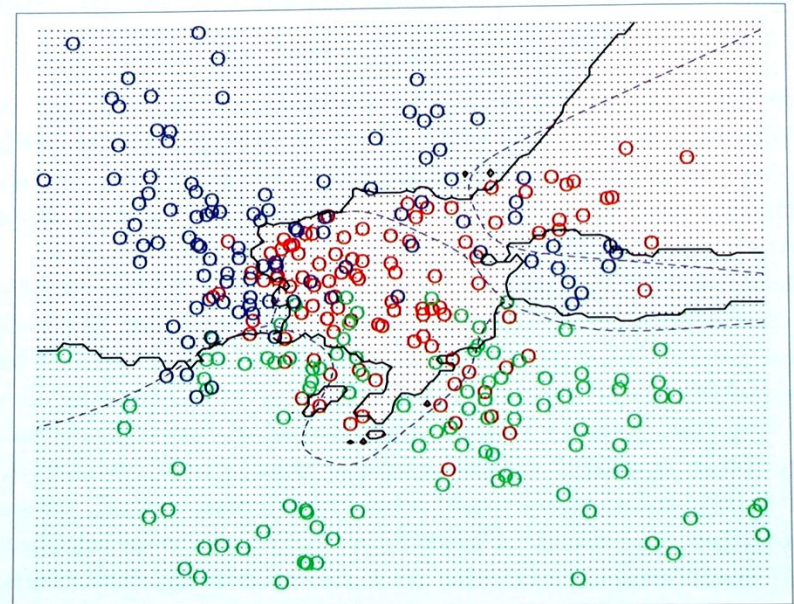


Example: Effect of k

1-Nearest Neighbor



15-Nearest Neighbors



Supervised Learning

- 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-NN 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: $y_i \in \mathfrak{R}$
- Parameter:
 - Similarity function: $K : X \times X \rightarrow \mathfrak{R}$
 - Number of nearest neighbors to consider: k
- Prediction rule
 - New example x'
 - K-nearest neighbors: k train examples with largest $K(\vec{x}_i, \vec{x}')$

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

Collaborative Filtering

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 options: "Watch Instantly", "Just for Kids", "Browse DVDs", "Your Queue", and "★ Suggestions For You". A search bar is located to the right of the menu.

The main content area features a 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". Below the poll are two rows of radio buttons for "Goofy" and "Raunchy".

Three movie posters are displayed with star ratings below them:

- Pulling**: A poster showing a group of people sitting on a couch. Rating: 5 stars.
- high life**: A poster featuring a man in a brown jacket. Rating: 5 stars.
- LEAVES OF GRASS**: A poster with a man and a woman. Rating: 5 stars.

At the bottom, there are two sections: "Recently Watched" and "Top 10 for Thorsten".

Recently Watched includes:

- TRAILER PARK BOYS**: A poster showing a man's belly with a tattoo.

Top 10 for Thorsten includes:

- THE LAST ENEMY**: A poster with a man's face.
- GEORGE GENTLY**: A poster with two men in trench coats.
- MI-5**: A poster with several men in suits.
- LOVE THE BEAST!**: A poster with a man in a red jacket.