Instance-Based Learning

CS4780/5780 – Machine Learning
Fall 2019

Nika Haghtalab & Thorsten Joachims
Cornell University

Reading: UML 19.1, 19.3
Optional Reading: Linden et al., Amazon Recommendations
(http://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf)

Supervised Learning

- Supervised Learning for Binary Classification:
  Acquire an operational classification rule given positive and negative training examples.

Also called: concept learning,...

Binary Classification Example

	correct (complete, partial, guessing)	color (yes, no)	original (yes, no)	presentation (clear, unclear)	latex (yes, no)	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

Instance Space X: Set of all possible instances x describable by attributes (often called features).

Target Attribute Y: Label $y \in \{+1, -1\}$ (or yes/no, or 0/1) for each instance.

Target Function f: Function that assigns true label for each x (f is unknown).

Example (x, y): Instance x with label $y = f(x)$.

Training Data S: Collection of examples observed by learning algorithm.

Learning a Binary Function

- Task:
  - Learn (to imitate) a function $f\colon X \quad \{+1,-1\}$
- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs → 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 $h\colon X$ ⍰ $\{+1,-1\}$ that approximates $f\colon X \quad \{+1,-1\}$ as well as possible.

K-Nearest Neighbor (KNN)

- Given: Training data $((\vec{x}_1, y_1), \ldots, (\vec{\mathrm{x}}_\mathrm{n}, y_\mathrm{n}))$
  - Attribute vectors: $\vec{x}_i \in X$
  - Labels: $y_i \in Y$
- Parameter:
  - Similarity function: $K : X \times X \rightarrow \Re$
  - 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 (complete, partial, guessing)	color (yes, no)	original (yes, no)	presentation (clear, unclear)	latex (yes, no)	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

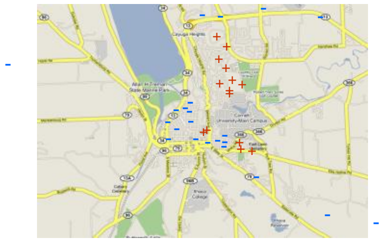
- Given: Training datadata $\left((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\right)$
  - Attribute vectors: $\vec{x}_i \in X$
  - Target attribute: $y_i \in Y$
- Parameter:
  - Similarity function: $K : X \times X \rightarrow \Re$
  - Number of nearest neighbors to consider: k
- Prediction rule
  - New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

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

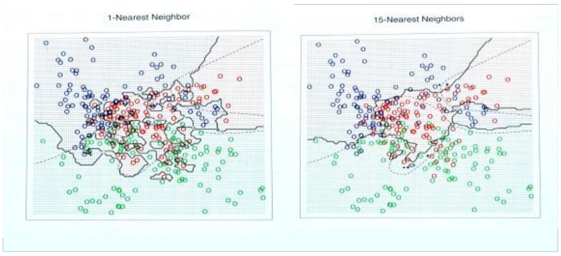
Types of Attributes

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

Example: Expensive Housing (>$200 / sqft)



Example: Effect of k



Hastie, Tibshirani, Friedman 2001

Supervised Learning

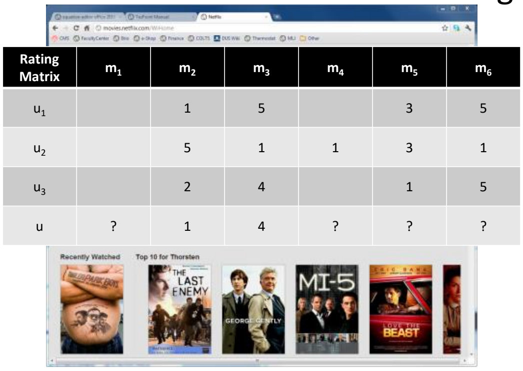
- Task:
  - Learn (to imitate) a function f: X □ Y
- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs → training examples
  - An example is a pair $(x, f(x))$, where x is the input and $f(x)$ is the output of the function applied to x.
- Goal: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

Weighted K-NN for Regression

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

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

Collaborative Filtering



Rating Matrix	m_1	m_2	m_3	m_4	m_5	m_6
u_1		1	5		3	5
u_2		5	1	1	3	1
u_3		2	4		1	5
u	?	1	4	?	?	?