Lazy and Eager Learning

Lazy: wait for query before generalizing
- \( k \)-Nearest Neighbor, Case based reasoning

Eager: generalize before seeing query
- Radial basis function networks, ID3, Backpropagation, NaiveBayes, ...
Instance-Based Learning

Key idea: just store all training examples \(<x_i, f(x_i)>\)

Nearest neighbor:
- Given query instance \(x_q\), first locate nearest training example \(x_n\), then estimate \(f(x_q) \leftarrow f(x_n)\)

Problem of noisy labels?

Adding Robustness

\(k\)-Nearest neighbor method:
- Given \(x_q\), take vote among its \(k\) nearest neighbors (if discrete-valued target function)
- Take mean of \(f\) values of \(k\) nearest neighbors (if real-valued)

\[
f(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_n)}{k}
\]
- Take majority for discrete classes

When To Consider kNN

- Instances map to points in \(\mathbb{R}^n\)
- Fewer than 20 attributes per instance
- Lots of training data

Advantages:
- Training is very fast
- Learn complex target functions
- Don't lose information

Disadvantages:
- Slow at query time
- Easily fooled by irrelevant attributes
**Voronoi Diagram**

Partition of space by nearness to instances.

**Distance-Weighted $k$NN**

Might want weight nearer neighbors more heavily...

$$f(x_q) \leftarrow \Sigma_{i=1}^k w_i f(x_{i,q}) / \Sigma_{i=1}^k w_i$$

where $w_i \equiv 1/d(x_q, x_i)^2$

and $d(x_q, x_i)$ is distance between $x_q$ and $x_i$

Note now it makes sense to use *all* training examples instead of just $k$
Curse of Dimensionality

Imagine instances described by 20 attributes, but only 2 are relevant to target function.

*Curse of dimensionality:* NN is easily misled in high-dimensional space.

How do data requirements grow with dimensionality?

Attribute Weighting:

- Stretch $j$th axis by weight $z_j$, where $z_1, \ldots, z_n$ chosen to minimize prediction error.
- Use cross-validation to automatically choose weights $z_1, \ldots, z_n$.
- Note setting $z_j$ to zero eliminates this dimension altogether.

See Moore and Lee (1994).

Lazy Learning

IBL Advantages:

- Learning is trivial
- Works
- Noise Resistant
- Rich Representation, Arbitrary Decision Surfaces
- Easy to understand

Disadvantages:

- Need lots of data
- Computational cost: memory, expensive application time
- Restricted to $x \in \mathbb{R}^n$
- Implicit weights of attributes (need normalization)
Case-Based Reasoning

Can apply instance-based learning even when $X \subseteq \mathbb{R}^n$
- need different “distance” metric

*Case-Based Reasoning* is instance-based learning applied to instances with symbolic logic descriptions

Example:
- (user-complaint error53-on-shutdown)
  - (cpu-model PowerPC)
  - (operating-system Windows)
  - (network-connection PCIA)
  - (memory 48meg)
  - (installed-applications Excel Netscape VirusScan)
  - (disk 1gig)
  - (likely-cause ???)

CBR in CADET

*CADET*: 75 stored examples of mechanical devices
- each training example: < qualitative function, mechanical structure >
- new query: desired function,
- target value: mechanical structure for this function

Distance metric: match qualitative function descriptions
CBR in CADET

A stored case:  T-junction pipe

Structure:

\[ Q_1, T_1 \]
\[ Q_2, T_2 \]
\[ Q_3, T_3 \]

Function:

\[ Q_1 \rightarrow Q_3 \]
\[ Q_2 \]
\[ T_1 \rightarrow T_3 \]
\[ T_2 \]

A problem specification:  Water faucet

Structure:

? 

Function:

\[ Q_c \]
\[ Q_h \]
\[ Q_{eq} \]
\[ T_c \]
\[ T_h \]
\[ T_{eq} \]

Bottom line:

- Simple matching of cases useful for tasks such as answering help-desk queries
- Area of ongoing research
Sources

- ML: 8.1, 8.2, 8.5
- Slides by Tom Mitchell as provided by Michael Littman