Computational learning and discovery

CSI 873 / MATH 689

Instructor: I. Griva

Wednesday 7:20 - 10 pm

Instance-Based Learning

- k-Nearest Neighbor
- Locally weighted regression
- Radial basis functions

Key idea: just store all training examples $\langle x_i, f(x_i) \rangle$

Key advantage: local estimation of a target function. It works well when a target function is complex but can be described by a collection of less complex local approximations

k - nearest neighbor

Nearest neighbor:

- Given query instance x_q, first locate nearest training example x_n, then estimate $\hat{f}(x_q) \leftarrow f(x_n)$

k-Nearest neighbor:

- Given x_q, take vote among its k nearest nbrs (if discrete-valued target function)
- take mean of f values of k nearest nbrs (if real-valued)

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

k - nearest neighbor

When To Consider Nearest Neighbor

- Instances map to points in $\Re^n$
- Small number of attributes
- Lots of training data

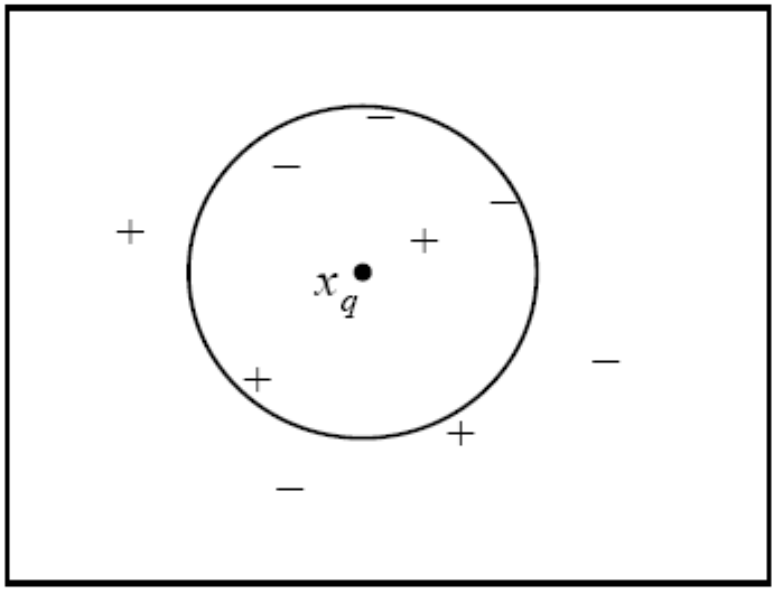
Advantages:

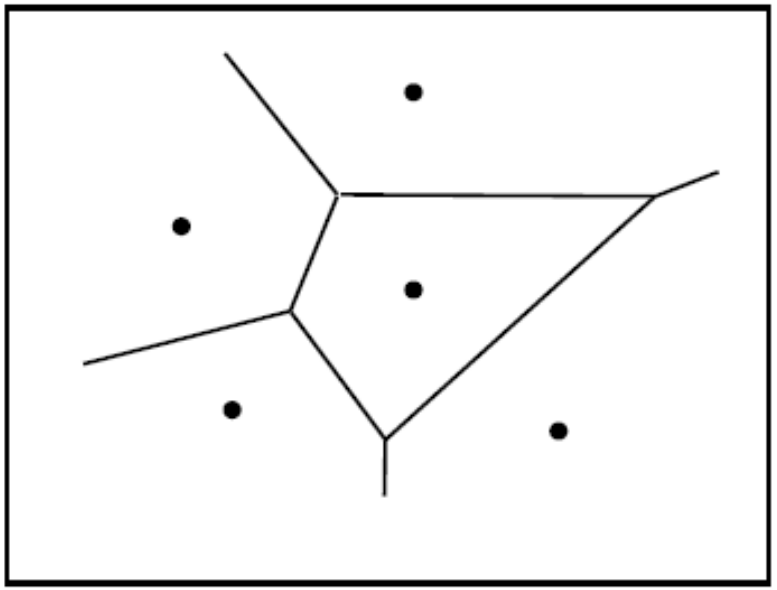
- Training is very fast
- Learn complex target functions
- Don't lose information

Disadvantages:

- Slow at query time
- Easily fooled by irrelevant attributes

k - nearest neighbor





Voronoi Diagram

Distance weighted *k* - nearest neighbor

Might want weight nearer neighbors more heavily...

$$\hat{f}(x_q) \leftarrow \frac{\Sigma_{i=1}^{k} w_i f(x_i)}{\Sigma_{i=1}^{k} w_i}$$

where

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

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

k - nearest neighbor

Curse of Dimensionality

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

Curse of dimensionality: nearest nbr is easily mislead when high-dimensional X

One approach:

- Stretch jth 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

Locally Weighted Regression

Note kNN forms local approximation to f for each query point x_q

Why not form an explicit approximation $\hat{f}(x)$ for region surrounding x_q

Locally Weighted Regression

1. Minimize the squared error over just the k nearest neighbors:

$$E_1(x_q) \equiv \frac{1}{2} \sum_{x \in k \text{ nearest nbrs of } x_q} (f(x) - \hat{f}(x))^2$$

2. Minimize the squared error over the entire set D of training examples, while weighting the error of each training example by some decreasing function K of its distance from x_q:

$$E_2(x_q) \equiv \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2 \; K(d(x_q, x))$$

3. Combine 1 and 2:

$$E_3(x_q) \equiv \frac{1}{2} \sum_{x \in k \text{ nearest nbrs of } x_q} (f(x) - \hat{f}(x))^2 \; K(d(x_q, x))$$

ANN

$$E \equiv \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

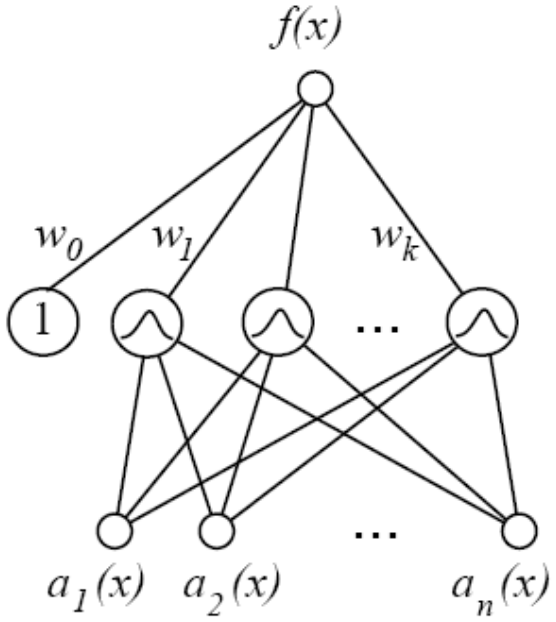
Radial Basis Functions

- Global approximation to target function, in terms of linear combination of local approximations
- Used, e.g., for image classification
- A different kind of neural network
- Closely related to distance-weighted regression, but "eager" instead of "lazy"

Advantages

Can be trained more efficiently than feed forward networks trained with Backpropagation, because the input layer and the output layer of an RBF are trained separately.

Radial Basis Functions



where $a_i(x)$ are the attributes describing instance x, and

$$f(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x))$$

One common choice for $K_u(d(x_u, x))$ is

$$K_u(d(x_u, x)) = e^{-\frac{1}{2\sigma_u^2} d^2(x_u, x)}$$

Training Radial Basis Functions

Q1: What x_u to use for each kernel function $K_u(d(x_u, x))$

- Scatter uniformly throughout instance space
- Or use training instances (reflects instance distribution)

Q2: How to train weights (assume here Gaussian K_u)

- First choose variance (and perhaps mean) for each K_u
  - e.g., use EM
- Then hold K_u fixed, and train linear output layer
  - efficient methods to fit linear function

Case Based Reasoning

CADET: 75 stored examples of mechanical devices

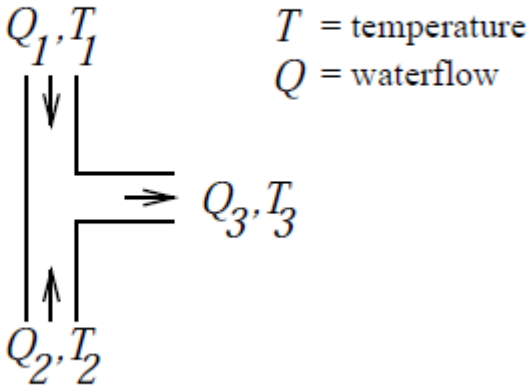
- each training example: ⟨ qualitative function, mechanical structure⟩
- new query: desired function,
- target value: mechanical structure for this function

Distance metric: match qualitative function descriptions

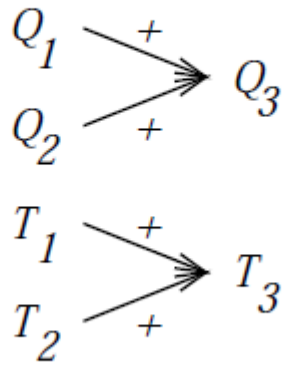
Case Based Reasoning

A stored case: T−junction pipe

Structure:



Function:

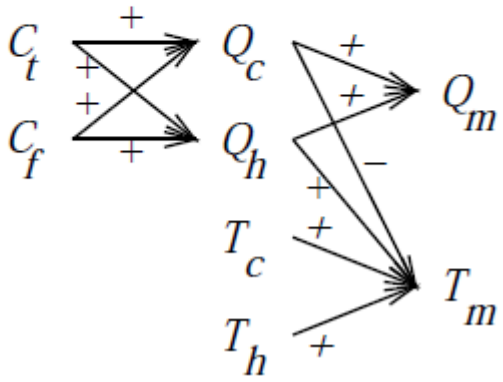


A problem specification: Water faucet

Structure:

?

Function:



Lazy vs. Eager Learner

- Eager learner must create global approximation
- Lazy learner can create many local approximations
- if they use same H, lazy can represent more complex fns (e.g., consider H = linear functions)