Machines Learning Part 2
Instance-based learning

With elements inspired by: Raymond J. Mooney

Philippe Pasquier
Office 565 (floor 14)
pasquier@sfu.ca

SFU

Philippe Pasquier, January 2008

Introduction

- **Instance-based learning techniques (aka case-based or memory-based or non-parametric) construc a general, explicit description of the target function by simply storing the training examples.**
- **It is a family of techniques:**
  - **Nearest neighbor**
  - **Locally weighted regression**

    } Instances are points in an Euclidean space
  - **Symbolic representation techniques**
- **Refered to as lazy learning because the processing is delayed until a new instance must be classified.**
- **The advantage is that it estimate the target function locally and differently for each instance rather than for the entire space**

Similarity / Distance Metrics

- **Instance-based methods assume a function for determining the similarity or distance between any two instances.**
- **For continuous feature vectors, Euclidian distance is the generic choice:**

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^{n} (a_p(x_i) - a_p(x_j))^2}$$

Where $a_p(x)$ is the value of the pth feature of instance x.

- To compensate for difference in units across features, scale all continuous values to the interval [0,1].
- For discrete features, we assume that the distance between two values is 0 if they are the same and 1 if they are different.
- **Other distance metrics are available: Mahalanobis distance, Cosine Similarity, Pearson correlation, Edit distance, ...**

K-Nearest Neighbor Algorithm

- **Training algorythm:**
  - **Memorize each training example** $< x, f(x) >$
- **Classification algorythm:**
  - **Given the request,** x_q **:**
    - **Calculate the distance between** x_q **and each training example** x **:** $d(x_q, x)$
    - **Select** $x_{1,} \ldots, x_k$**, the k closest instances to** x
    - **Return the estimation of** $f(x_q)$**:**

$$\underset{v \in V}{argmax} \sum_{i=1}^{k} \delta(v, f(x_i))$$

Where $\delta(v, f(x_i))=1$ if $v=f(x_i)$ and $\delta(v, f(x_i))=0$ otherwise

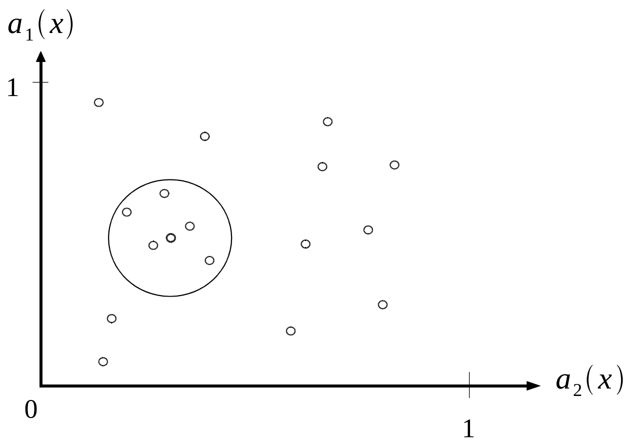
Using K-Nearest Neighbor

- **This version is for discrete valued target functions:** $f : \mathbb{R}^n \rightarrow V$, where $V = \{v_{1,} \dots , v_s\}$
- **Usually use odd value for *k* to avoid ties.**
- **But we can do a version for real-valued target functions** $f : \mathbb{R}^n \rightarrow \mathbb{R}$ **by returning:**

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

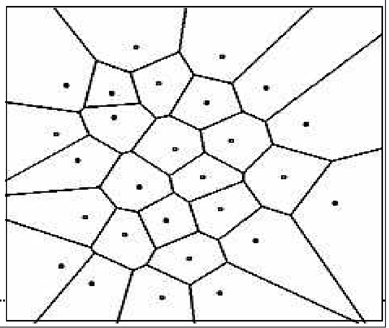
- **i.e: Pick the *k* closest training examples and assign their main value to the test instance**

5-Nearest Neighbor Example



Implicit Classification Function

- Although it is not necessary to explicitly calculate it, the learned classification rule is based on regions of the feature space closest to each training example.
- For 1-nearest neighbor with Euclidian distance, the Voronoi diagram gives the complex polyhedra segmenting the space into the regions closest to each point.
- Why k>1?
  Voting multiple neighbors helps decrease susceptibility to noise.

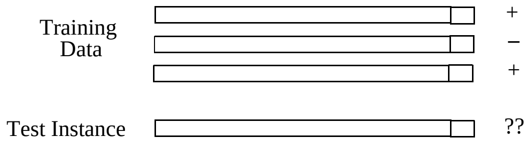


Computational Performance

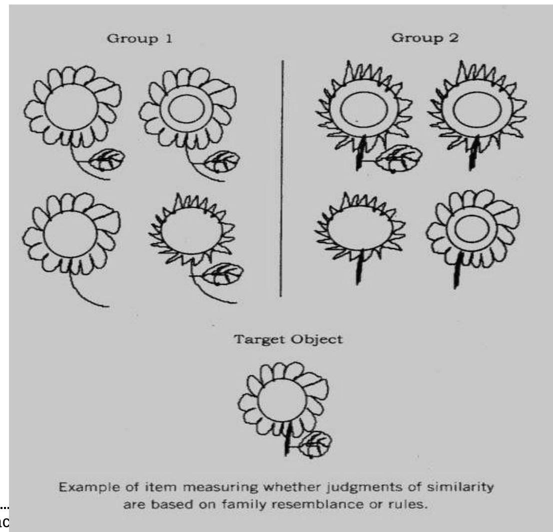
- Linear search to find the nearest neighbors is not efficient for large training sets. That is a lot of distances to compute!
- Indexing structures can be built to speed testing:
  – For Euclidian distance, a kd-tree can be built that reduces the expected time to find the nearest neighbor to O(log *n*) in the number of training examples.
    - Nodes branch on threshold tests on individual features and leaves terminate at nearest neighbors.
  – Other indexing structures possible for other metrics or string data.
    - Inverted index for text retrieval.

Feature Relevance and Weighting

- Standard distance metrics weight each feature equally when determining similarity.
  - Problematic if many features are irrelevant, since similarity along many irrelevant examples could mislead the classification.
  - Solution: Features can be weighted by some measure that indicates their ability to discriminate the category of an example, such as information gain.
- Overall, instance-based methods favor global similarity.



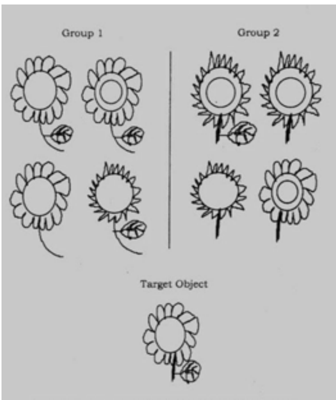
Instances in Human Learning Biases



Example of item measuring whether judgments of similarity are based on family resemblance or rules.

Instances in Human Learning Biases

- Psychological experiments show that people from different cultures exhibit distinct categorization biases.
- "Western" subjects favor simple rules (straight stem) and classify the target object in group 2.
- "Asian" subjects favor global similarity and classify the target object in group 1.



Example of item measuring whether judgments of similarity are based on family resemblance or rules.

Nearest Neighbor Variations

- Many variations are possible:
  - Distance Weighted nearest neighbor algorithm:
    - The idea is to weight the contribution of each of the k-neighbors according to their distance (see example before), giving a greater weight to closer neighbors.
    - All training examples can be used to help classify a test instance by giving every training example a vote that is weighted by the inverse square of its distance from the test instance.
  - Relevance weighted version:
    - Streching axes in the Euclidean space according to their relevance:
      - Shortening axes that correspond to irrelevant attributes (factor 0 suppress them)
      - Lenghtening axes that correspond to relevant ones

Conclusion

- IBL methods classify test instances based on similarity to specific training instances rather than forming explicit generalizations.
- Typically trade decreased training time for increased testing time.
- K-nearest neighbor is a highly effective inductive inference method for many practical problems
- Other methods include:
  - Locally weighted regression
  - Radial basis functions
  - Case-based reasoning