

Lecture 4

Nearest Neighbors

Curse of Dimensionality (take 2)

last time we saw one approach to curse of dimensionality: assume conditional independence of data dimensions given class, model each class-conditional distribution separately

each distribution has associated parameters (Bernoulli probabilities; Gaussian means and variances): *parametric* data model

another possibility is to build a *non-parametric* data model, without assuming known form of underlying density

embodies often sensible assumption about data:

1. data occupies sub-space of high-dimensional input space
2. output (class) varies smoothly with input

Instance-Based Learning

simple methods for approximating discrete-valued or real-valued target functions (classification or regression problems)

learning amounts to simply storing training data

test instances classified using similar training instances

most basic instance-based method: *nearest neighbor* algorithm

Nearest Neighbors

assumes instances correspond to points in d -dimensional Euclidean space

target function value for new query estimated from known value of nearest training example

distance typically defined to be Euclidean:

$$\|\mathbf{a} - \mathbf{b}\| = \sqrt{\sum_{j=1}^d (a_j - b_j)^2}$$

algorithm: find example $\langle \mathbf{x}^*, c(\mathbf{x}^*) \rangle$ closest to test instance \mathbf{x}_q

output: $\hat{c}(\mathbf{x}_q) = c(\mathbf{x}^*)$

note: need not compute square-root (same answer without it)

Decision Boundaries: Voronoi diagram

Nearest neighbor algorithm does not explicitly compute *decision boundaries*, but these can be inferred

decision boundaries

- show how input space divided into classes
- form a subset of the Voronoi diagram for the training data
- each line segment is equidistant between two points of opposite classes

complex models \Rightarrow possibly complicated decision boundaries

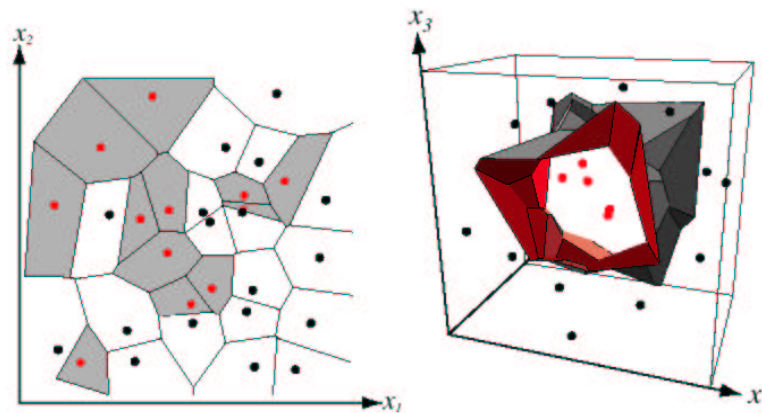


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Nearest Neighbors: Problems & Remedies

1. sensitive to mis-labeled data (“class noise”) \Rightarrow smooth by having k nearest neighbors vote

algorithm: find k examples $\{ \langle \mathbf{x}_1, c(\mathbf{x}_1) \rangle; \dots; \langle \mathbf{x}_k, c(\mathbf{x}_k) \rangle \}$ closest to test instance \mathbf{x}_q

classification output (majority):

$$\hat{c}(\mathbf{x}_q) = \arg \max_{c_z \in C} \sum_{r=1}^K \delta(c_z, c(\mathbf{x}_r))$$

2. some attributes have larger ranges, are treated as more important \Rightarrow normalize scale
3. irrelevant, correlated attributes add noise to distance measure \Rightarrow eliminate some attributes, or vary and possibly adapt weight of attributes
4. non-metric attributes (symbols) \Rightarrow Hamming distance

Computational complexity, savings

brute-force approach to 1-NN – inspect each stored point, calculate Euclidean distance to test point, keep closest: $O(dn^2)$

reduce computational burden:

1. use subset of dimensions
2. use subset of examples
 - form efficient search tree from examples (kd-tree)
 - remove examples that lie within Voronoi region

Nearest Neighbors: Advantages

1. retains all information in training instances
2. can approximate complex target functions – only using simple local approximations
3. need not pre-classify entire input space
4. learning can be on-line or batch

Nearest Neighbors: Summary

both k -NN and naive Bayes build $P(c_k|\mathbf{x})$

key assumption of k -NN: smoothness (property of input point \mathbf{x} likely to be similar to those of points in neighborhood of \mathbf{x})

key decision: how to define neighborhood (need to contain some points but not all)

alternative approach: *Parzen windows* builds fixed size neighborhoods as opposed to fixed number of neighbors

k -NN can work for densely or sparsely populated regions

choice of distance metric important

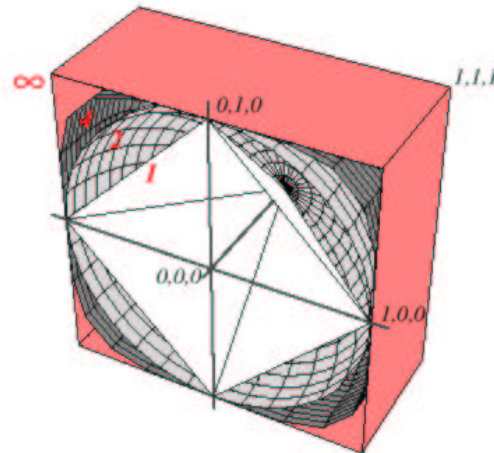


FIGURE 4.19. Each colored surface consists of points a distance 1.0 from the origin, measured using different values for k in the Minkowski metric (k is printed in red). Thus the white surfaces correspond to the L_1 norm (Manhattan distance), the light gray sphere corresponds to the L_2 norm (Euclidean distance), the dark gray ones correspond to the L_4 norm, and the pink box corresponds to the L_∞ norm. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

general “ L_k ” norm (Minkowski metric):

$$L_k(\mathbf{a}, \mathbf{b}) = \left(\sum_{j=1}^d |a_j - b_j|^k \right)^{1/k}$$

but problems with dimensionality: neighbors in high-dimensional spaces are far away