Prototype Methods and Nearest Neighbor

Henrik I. Christensen

Robotics & Intelligent Machines @ GT
Georgia Institute of Technology,
Atlanta, GA 30332-0280
hic@robotics.gatech.edu

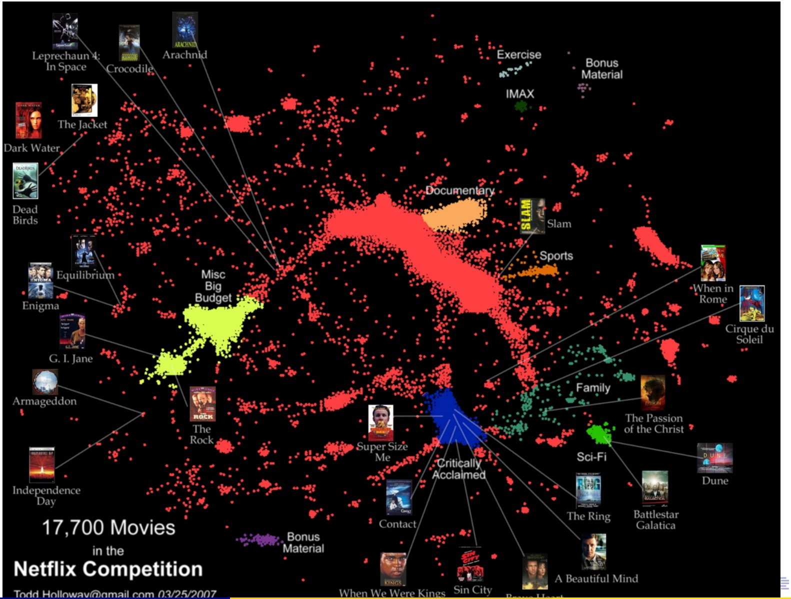
Outline

1. Introduction
2. Prototype Methods
3. k-Nearest Neighbor Classifier
4. Adaptive Nearest Neighbor Classifier
5. Summary

Introduction

- Sometimes it is easier to use the data directly or in a simplified form
- Data structure might be hard to parse
- The disadvantage of “raw” data models is the lack of insight
- Analysis of robustness / performance can be a challenge
- Yet, often these methods are very effective

Netflix Example



17,700 Movies

in the

Netflix Competition

Todd.Holloway@gmail.com 03/25/2007

Outline

1. Introduction
2. Prototype Methods
3. k-Nearest Neighbor Classifier
4. Adaptive Nearest Neighbor Classifier
5. Summary

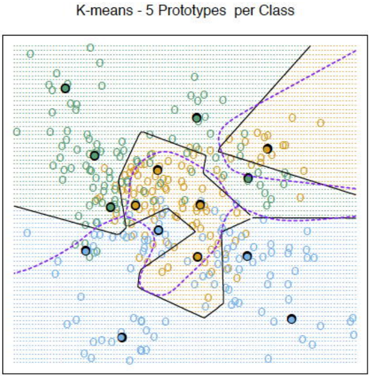
K-means

- Assume you have a collection of data $\{x_1, \ldots, x_n\}$
- We want to approimate the data by K "prototypes"
- Generate an initial guess of K prototypes
- Iterate to convergence
  1. For each data member (x_i) find closest "prototype"
  2. Re-estimate the center for the cluster
- For any data-member approximate it by its mean (thus, the K-means)
- So far without consideration of classes

K-means with classes

- Assume we have R classes.
- Each prototype is represented by (x_i, g_j), where x_i is the data value and g_j is the label
- Apply K-means to each class of data separately
- Assign class labels to each of the K*R prototypes
- Assign class label to new data based on nearest neighbor

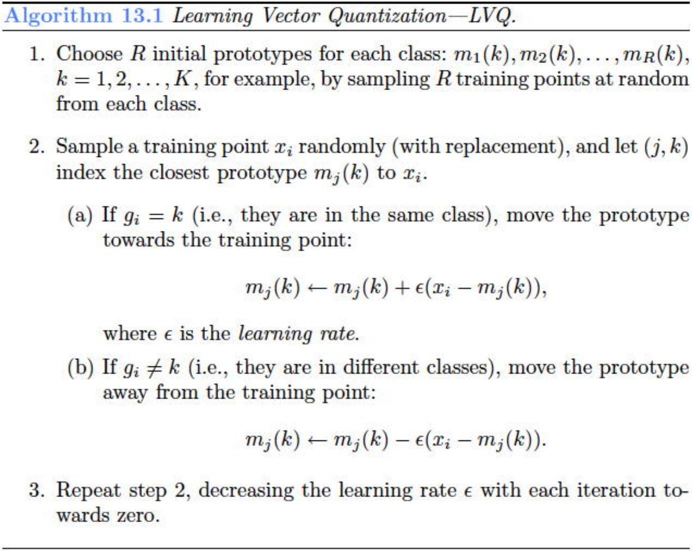
Example K-means result



Fixing K-means

- The prototypes are generated for each class independently
- The boundaries may not be well-defined
- What if we could change this as part of learning?

Kohonen's Learned Vector Quantization



Algorithm 13.1 *Learning Vector Quantization—LVQ.*

1. Choose R initial prototypes for each class: $m_1(k), m_2(k), \ldots, m_R(k)$, $k = 1, 2, \ldots, K$, for example, by sampling R training points at random from each class.

2. Sample a training point x_i randomly (with replacement), and let (j, k) index the closest prototype $m_j(k)$ to x_i.

   (a) If $g_i = k$ (i.e., they are in the same class), move the prototype towards the training point:

   $$m_j(k) \leftarrow m_j(k) + \epsilon(x_i - m_j(k)),$$

   where ϵ is the *learning rate*.

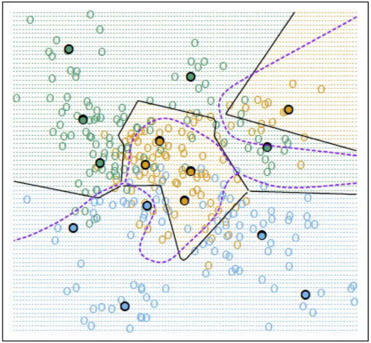
   (b) If $g_i \neq k$ (i.e., they are in different classes), move the prototype away from the training point:

   $$m_j(k) \leftarrow m_j(k) - \epsilon(x_i - m_j(k)).$$

3. Repeat step 2, decreasing the learning rate ϵ with each iteration towards zero.

Example LVQ result

LVQ - 5 Prototypes per Class



Gaussian Mixture Models

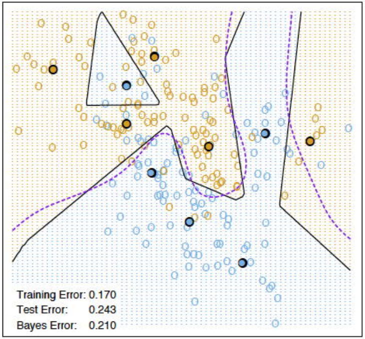
- K-means is a hard method for approximating data
- Could we use a mixture of Gaussians to approxiate our data / classes
- Model each class k as

$$P(X|G = k) = \sum_{r=1}^{K} \pi_{kr}\phi(X; \mu_{kr}, \Sigma)$$

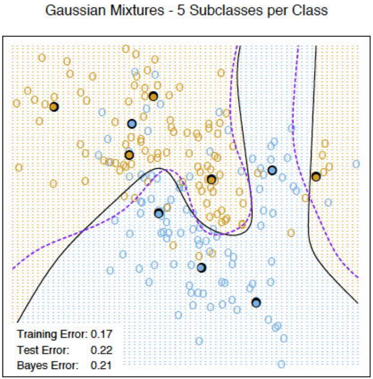
- The GM model is typically more robust to noise

Example K-means result

K-means - 5 Prototypes per Class



Example Gaussian Mixture result



Outline

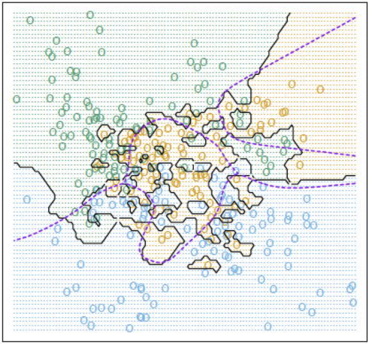
1. Introduction
2. Prototype Methods
3. k-Nearest Neighbor Classifier
4. Adaptive Nearest Neighbor Classifier
5. Summary

K-Nearest Neighbors Classifier

- Find the k nearest neighbors $\{(x_1, g_1), \dots, (x_k, g_k)\}$
- Estimate the class by majority vote
- In most cases a simple Euclidian distance is used
- This is a pure memory based technique. All training data are preserved
- It is possible to show that the error rate at most is twice the Bayes error rate (Ripley 1996).
- Still considered a top-10 data minign algorithm

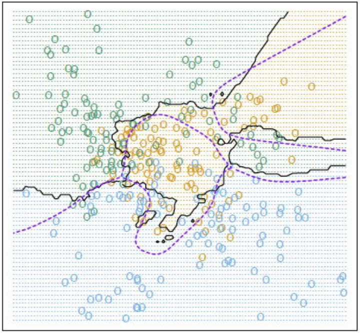
Example result - 1-NN

1-Nearest Neighbor



Example result - 15-NN

15-Nearest Neighbors



Real-world example - LANDSAT

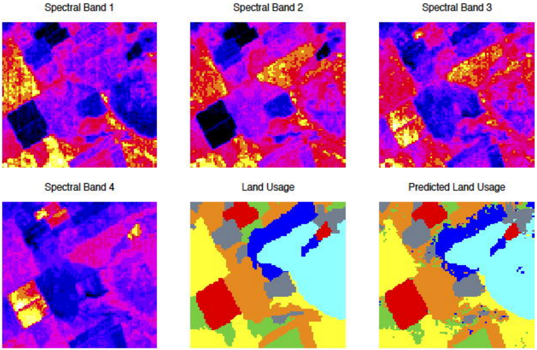


FIGURE 13.6. *The first four panels are LANDSAT images for an agricultural area in four spectral bands, depicted by heatmap shading. The remaining two panels give the actual land usage (color coded) and the predicted land usage using a five-nearest-neighbor rule described in the text.*

Real-world example - LANDSAT - Performance

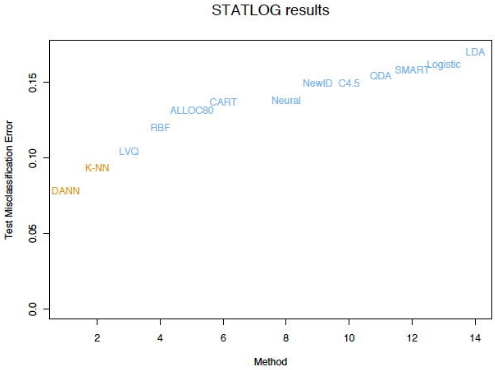


FIGURE 13.8. *Test-error performance for a number of classifiers, as reported by the STATLOG project. The entry DANN is a variant of k-nearest neighbors, using an adaptive metric (Section 13.4.2).*

Character recognition - MNIST

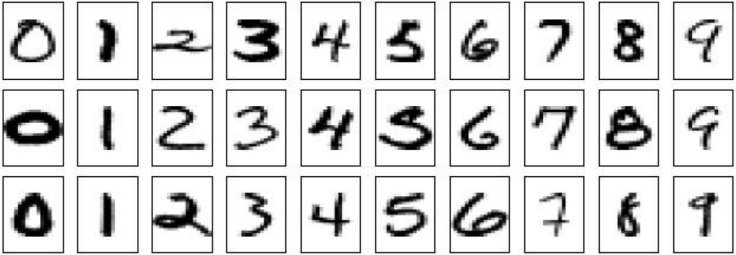
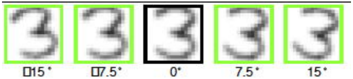


FIGURE 13.9. *Examples of grayscale images of handwritten digits.*

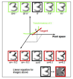
MNIST addressign systematic variations

- Consider slight variations in the rotation of the character

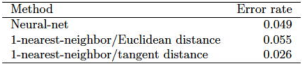


- The image is here 16×16 or 256 vector
- This is a curve in a 256D space.
- We could compute a curvature space and reduce dynamics

MNIST Curvature encoding



MNIST Curvature Comparative Results



Method	Error rate
Neural-net	0.049
1-nearest-neighbor/Euclidean distance	0.055
1-nearest-neighbor/tangent distance	0.026

K-NN considerations

- Classifying unknown data are relatively expensive
  - Have to compare / compute distances for k-neighbors
  - Computationally intensive, especially as size of training data grows
  - The challenge is particularly hard in high dimensional spaces
  - Noisy / irrelevant data can be a major challenge

Outline

1. Introduction

2. Prototype Methods

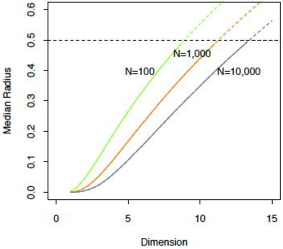
3. k-Nearest Neighbor Classifier

4. Adaptive Nearest Neighbor Classifier

5. Summary

Adaptive Neirest Neighbor Classifier

- The distance to a closeby point goes up quickly with higher dimensional spaces.
- Size considerations - the radius to find a point goes up quickly, ie the space coverage is sparse.



Discriminant adaptive neighbor classification

- DANN
  - Discriminative - senstitive the set of classes
  - Adaptive - capability to adapt / adjust ot the situation
  - NN - based on the local neighbors
- Uses local discriminative analysis to determine the right neighborhood

The DANN algorithm

1. Initialize Σ to be identity I
2. Given a test point x_0 find nearest neighbor using the metric

$$D(x, x_0) = (x - x_0)^T \Sigma (x - x_0)$$

compute the weighted within, W, and between, B covariances

- Update the Σ matrix using the metric

$$\begin{aligned} \Sigma &= W^{-1/2}[W^{-1/2}BW^{-1/2} + \epsilon I]W^{-1/2} \\ &= W^{-1/2}[B^* + \epsilon I]W^{-1/2} \end{aligned}$$

- Iterate 1-3 a number of times to find the adjusted nearest neighbors

basic example of DANN use

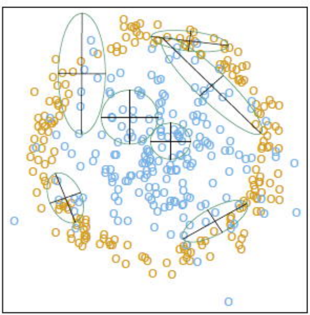


FIGURE 13.14. *Neighborhoods found by the DANN procedure, at various query points (centers of the crosses). There are two classes in the data, with one class surrounding the other. 50 nearest-neighbors were used to estimate the local metrics. Shown are the resulting metrics used to form 15-nearest-neighborhoods.*

Outline

1. Introduction

2. Prototype Methods

3. k-Nearest Neighbor Classifier

4. Adaptive Nearest Neighbor Classifier

5. Summary

Summary

- Prototype / Memory Based Techniques Frequently perform well, especially on unstructured data
- Computational considerations are important
- Often k-NN or basic mixture models are good for a first evaluation of performance
- Lots of good tools available for use even on large data sets.