Discriminant Adaptive Nearest Neighbor Classification (DANN)

Soo-Young Kim

November 10, 2015

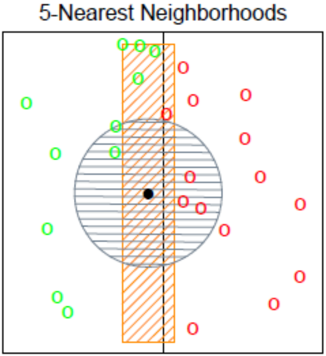

FIGURE: The points are uniform in the cube, with the vertical line separating class red and green. The vertical strip denotes the 5-nearest-neighbor region using only the horizontal coordinate to find the nearest-neighbors for the target point (solid dot). The sphere shows the 5-nearest-neighbor region using both coordinates, and we see in this case it has extended into the class-red region (and is dominated by the wrong class in this instance).(ESL)

DANN

- K-nearest Neighbor method has problems in high dimensional data set

Discriminant Adaptive Nearest Neighbor Classification(DANN)

- DANN uses local linear discriminant analysis to estimate an effective metric for computing neighborhoods

- DANN utilizes a small tuning parameter to shrink or stretch neighborhoods (The neighborhoods stretch out in directions for which the class probabilities don't change much.)

Setup

Consider a discrimination problem with

- N training observations from J classes
- Training data: $\boldsymbol{x} = (x_1, \cdots, x_p)$ and known class memberships
- Test point : $\boldsymbol{x}_0$
- $\boldsymbol{B}$ and $\boldsymbol{W}$: the $p \times p$ between and within sum-of-squares matrices
- $\boldsymbol{\Sigma}$: a local metric

- Goal: Predict the class membership of an observation with predictor vector $\boldsymbol{x}_0$ as the most frequent class among the K neighbors

Setup

Tuning parameter

- K_M : the number of nearest neighbors in the neighborhood N_{K_M} for estimation of the metric (ex: $K_M = \max(N/5, 50)$)
- K: The number of neighbors in the final nearest neighbor rule (Note: larger K reduces variance/ small K reduces bias)
- ϵ: the "softening" parameter in the metric

Method

- Discriminant Adaptive Nearest Neighbor Classifier:

0) Initialize the metric $\boldsymbol{\Sigma} = \boldsymbol{I}$

1) Spread out a nearest neighborhood of K_M points around the test point $\boldsymbol{x}_0$, in the metric $\boldsymbol{\Sigma}$.

2) Calculate the weighted $\boldsymbol{W}$ and $\boldsymbol{B}$ using the points in the neighborhood

3) Define a new metric $\boldsymbol{\Sigma} = \boldsymbol{W}^{-1/2}[\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2} + \epsilon\boldsymbol{I}]\boldsymbol{W}^{-1/2}$

4) Iterate steps 1,2, and 3

5) At completion, use the metric $\boldsymbol{\Sigma}$ for K-nearest neighbor classification at the test point $\boldsymbol{x}_0$

- DANN metric: $D(\boldsymbol{x}, \boldsymbol{x}_0) = (\boldsymbol{x} - \boldsymbol{x}_0)'\boldsymbol{\Sigma}(\boldsymbol{x} - \boldsymbol{x}_0)$.

Details of the implementation

- A weight function at $\boldsymbol{x}_0$

$$w_i = k(\boldsymbol{x}_i, \boldsymbol{x}_0; \Sigma, h) = \phi_h(||\Sigma_0^{1/2}(\boldsymbol{x} - \boldsymbol{x}_0)||)$$

where ϕ_h is a symmetric function depending on a parameter h

- To determine metric $\boldsymbol{\Sigma} = \boldsymbol{W}^{-1/2}[\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2} + \epsilon\boldsymbol{I}]\boldsymbol{W}^{-1/2}$:

$$\boldsymbol{B}(\boldsymbol{x}_0; \boldsymbol{\Sigma}_0, h) = \sum_{j=1}^{J} \widehat{\pi}_j(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T,$$

$$\text{where } \widehat{\pi}_j = \frac{\sum_{y_i=j} w_i}{\sum_{i=1}^{N} w_i}$$

$$\boldsymbol{W}(\boldsymbol{x}_0; \boldsymbol{\Sigma}_0, h) = \sum_{j=1}^{J} \sum_{y_i=j} w_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T / \sum_{i=1}^{N} w_i$$

Example of DANN metric

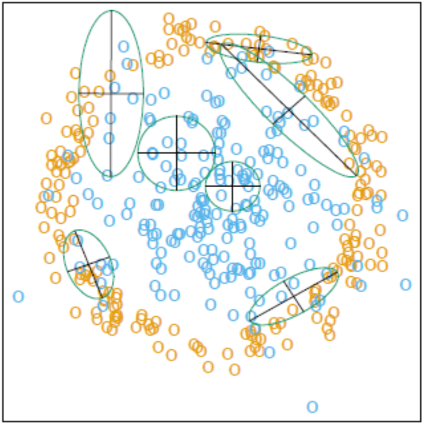

Figure: Neighborhoods found by the DANN procedure, at various query points (centers of the crosses). There are two classes in the data, with one class surrounding the other. 50 nearest-neighbors were used to estimate the local metrics. Shown are the resulting metrics used to form 15-nearest-neighborhoods.(ESL)

Remarks

- In the pure regions with only one class, the neighborhoods remain circular. ($\boldsymbol{B}$ =0 and $\boldsymbol{\Sigma} = I$)
- The ϵ parameter rounds the neighborhood, from an infinite strip to an ellipsoid, to avoid using points far away from x_0.
- When $\epsilon = 0$, the metric approximately behaves like LDA metric.
- In practice, it is more effective to estimate only the diagonal elements of $\boldsymbol{W}$, and off-diagonal elements are zero. (There might be insufficient data locally to estimate the $O(p^2)$ elemets)

Reference

Trevor Hastie and Robert Tibshirani. "Discriminant Adaptive Nearest Neighbor Classification." IEEE Trans. Pattern Anal. Mach. Intell. 18, 6 (June 1996), 607-616.