Distance Based Classification Algorithms

Nearest Neighbor Classifier(Cover and Hart, 1967)

- Let $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{n}$ be set of labeled patterns (Training set), where x_i is a pattern and y_i be its class label. Let x be a pattern with unknown class label (test pattern).
- NN Rule is :
- Let $x' \in \mathcal{T}$ be the pattern nearest to a test pattern x.
- $l(x) = l(x')$

Complexity:

- Time: $O(|\mathcal{T}|)$
- Space: $O(|\mathcal{T}|)$

Condensed NNC (Hart, 1968)

INPUT: Training Set $\mathcal{T}$
OUTPUT: A condenced Set $\mathcal{S}$.

1. Start with a condensed set $\mathcal{S} = \{x\}$.
2. For each $x \in \mathcal{T} \setminus \mathcal{S}$
    1. Classify x using NN considering $\mathcal{S}$ as training set.
    2. if x is misclassified then $\mathcal{S} = \mathcal{S} \cup \{x\}$
3. Repeat Step 2 until no change found in Condensed Set

Modified CNN(Devi and Murthy, 2003)

1. Start with condensed set $\mathcal{S}$. $\mathcal{S}$ contains one pattern from each class.
2. $\mathcal{G} = \emptyset$
3. For each $x \in \mathcal{T}$
   1. Classify x using NN considering $\mathcal{S}$ as training set.
   2. if x is misclassified then $\mathcal{G} = \mathcal{G} \cup \{x\}$
4. Find a representative pattern from each class in $\mathcal{G}$; Let representative set is $\mathcal{R}$.
5. $\mathcal{S} = \mathcal{S} \cup \mathcal{R}$
6. $\mathcal{G} = \emptyset$
7. Repeat Step 2 to Step 6 until there is no misclassification.

- MCNN is an order independent algorithm
- Many Iterations.

K-Nearest Neighbor Classifier

- Let $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{n}$ be a Training set.
- Let x be a pattern with unknown class label (test pattern).
- Algorithm:
- $KNN = \emptyset$
- For each $t \in \mathcal{T}$
  1. if $|KNN| <= K$
     - $KNN = KNN \cup \{t\}$
  2. else
     1. Find a $x' \in$ KNN such that $dis(x, x') > dis(x, t)$
     2. $KNN = KNN - \{x'\}$; $KNN = KNN \cup \{t\}$
- The pattern x belongs to a Class in which most of the patterns in KNN belong to.

How to find the value of K

r-fold Cross Validation

- Partition the training set into r blocks. Let these are $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_r$
- For $i = 1$ to r do
  1. Consider $\mathcal{T} - \mathcal{T}_i$ as the training set and $\mathcal{T}_i$ as the validation set.
  2. For a range of K values (say from 1 to m) find the error rates on the validation set.
  3. Let these error rates are $e_{i1}, e_{i2}, \ldots, e_{im}$
- Take e_i = mean of $\{e_{1i}, e_{2i}, \ldots, e_{ri}\}$, for $i = 1$ to m.
- The value of $K = \underset{i}{\text{argmin}}\{e_1, e_2, \ldots, e_m\}$

Weighted k-NNC(Dudani, 1976)

- *k-NNC gives equal importance to the first NN and to the last NN.*
- Weight is assigned to each nearest neighbor of a quary pattern.
- Let $\mathcal{X} = \{x_{i=1..k}^{C_j} \mid x_i^{C_j} \in \mathcal{T}\}$ be the set of k-NN of q, whose class label is to be determined.
- Let $D = \{d_1, d_2, \ldots, d_k\}$ be an ordered set, where $d_i = ||x_i - q||$, $d_i \leq d_j, i < j$
- The weight w_j is assigned to j^{th} nearest neighbor as follows.

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_j \neq d_1 \\ 1 & \text{if } d_j = d_1 \end{cases}$$

- Calculate weighted sums of patterns belong to each class.
- Classify q to a class for which weighted sum is maximum.

Editing Techniques

- Larger the training set, more the computational cost.
- Another technique eliminates (edit) training prototypes (pattern) erroneously labeled, commonly outliers, and at the same time, to "clean" the possible overlapping between regions of different classes.

Editing Techniques

- Larger the training set, more the computational cost.
- Another technique eliminates (edit) training prototypes (pattern) erroneously labeled, commonly outliers, and at the same time, to "clean" the possible overlapping between regions of different classes.
- Wilsons editing relies on the idea that, if a prototype is erroneously classified using the k-NN, it has to be eliminated from the training set.

Edited Nearest Neighbor (Willson, 1976)

INPUT: $\mathcal{T}$ is a training set
OUTPUT: S is an edited set

1. for each $x \in \mathcal{T}$ do
    1. classify x using k-NN Classifier (break the ties randomly)
    2. if x is misclassified then mark x
2. Delete all marked paterns from $\mathcal{T}$; let the reduced training set be S.
3. Output S

Repeated ENN (Tomek, 1976)

- Apply ENN method repeatedly until there is no chnage in the training set $\mathcal{T}$