Boosting Nearest Neighbor Classifiers for Multiclass Recognition

Vassilis Athitsos and Stan Sclaroff

Presented by Mohan Sridharan

K-Nearest Neighbors??

- Nearest Neighbor (KNN) classifiers popular for multi-class recognition – vision, pattern recognition.
- KNN approaches work well for multi-class problems, but need a distance measure.
- KNN sensitive to choice of distance measure, especially at higher dimensions.
- Solution: learn the best distance measure??

Boosting – what and why?

- We have seen boosting techniques used in several applications.
- Essentially train a bunch of weak learners on the data. Final classification is a ‘combination’ of the weak learners.
- Good for high dimensional data, but works best with binary decision problems.

KNN + Boosting??

- KNN good for multi-class problems but not great for high-dim data.
- Boosting great for high-dim data but ‘happier’ with binary problems.
- The combination of KNN and Boosting makes sense for high-dim multi-class data.
- Still faced with choice of distance measure, reducing multi-class problem to binary classification problem.

Contributions…

- Distance measure learned from data using boosting – linear weighted average of a set of distance measures.
- Reduction of multi-class classification problem to binary classification problem so that boosting can be used efficiently.

Triplet selection…

- Available: set of training samples of objects with class labels.
  \[(x, y(x)) \text{ m samples} \]
  \[x \in X, \ y \in Y\]
- Select triplet:
  \[(q,a,b), \ y(q) = y(a), y(q) \neq y(b) \]
  \[D(q,a) < D(q,b)\]
- Design classifiers based on distance measures for this set of triplets.
Associating Distances with Classifiers

- Define classifiers for every distance measure \( D \) on input dataset of objects:
  \[
  \begin{align*}
  D(q,a,b) &= D(q,b) - D(q,a) \\
  \overline{D}(q,a,b) &= \begin{cases} 
  1 & D(q,a) < D(q,b) \\
  0 & D(q,a) = D(q,b) \\
  -1 & D(q,a) > D(q,b)
  \end{cases}
  \end{align*}
  \]

- If \( \overline{D} \) correctly classifies all triplets, then \( D \) is a good measure for the corresponding KNN classifier.

Remember K in KNN

- Sufficient: simple majority in \( K \) closest neighbors.

- Sufficient: \( \overline{D} \) classifies correctly all triplets \( (q,a,b) \) such that \( a,b \) are among \( K \) nearest neighbors of \( q \) among objects of class \( y(a), y(b) \).

Learning Weighted Distance Measure

- Given: Training set of objects with class labels, Set of distance measures.
- Choose a set of triplets. Evaluate distance measures as weak learners – Generalized AdaBoost.
- Output a linear weighted combination of weak learners – linear weighted combination of distance measures.
  \[
  H_i = \sum_{j=1}^{n} a_j D_j
  \]
  \[
  D'_w(x_1,x_2) = \sum_{j=1}^{n} a_j D_j(x_1,x_2)
  \]
  \[
  D'_w = H_i
  \]

Iterative Refinement

- \( T(D, r) = \) set of triples \( (q,a,b) \):
  - \( q \) = object from the available training set.
  - \( a \) = same class \( r \)-th nearest neighbor.
  - \( b \) = \( w \)-class \( r \)-th nearest neighbor \( w \neq y(q) \).
- \( T'(D, r) = \) union over \( T(D, r) \).
  - Selection of \( r \) based on knowledge of ‘k’ in KNN...
- Sample from \( T(D, r_{max}) \) and iterate until termination condition – whiteboard??
- Theoretical considerations, Computational complexity...

Observations...

- Tested on 8 UCI datasets, including 3 visual datasets.
- Compared with AdaBoost (w/o distance measure learning) and Naïve KNN.
- No clear winner – the paper accepts this!
- Each algorithm works well for some datasets – current one does worse for the segmentation dataset ☹

Observations...

- Comparable performance with established algorithms – worth further analysis??
- No Convergence guarantees – future work...
- Issues of scaling to high-dim and larger samples.
That's all folks 😊