Curse of Dimensionality

• Problems of high dimensional data, “the curse of dimensionality”
  – Running time
  – Overfitting
  – Number of samples required
• Affects most of the machine learning approaches
Curse of Dimensionality in k-NN

As number of dimensions increases, distance between points becomes larger, “neighborhood” become large.

If number of relevant attributes is fixed, increasing the number of less relevant attributes may swamp distance. Distance become less reliable.

\[
D(c_1, c_2) = \sqrt{\sum_{i=1}^{\text{relevant}} (\text{attr}_i(c_1) - \text{attr}_i(c_2))^2 + \sum_{j=1}^{\text{irrelevant}} (\text{attr}_j(c_1) - \text{attr}_j(c_2))^2}
\]

Curse of Dimensionality in k-NN

• Suppose we have 5000 points uniformly distributed in the unit hypercube and we want to apply 5-NN. Suppose our query point is at the origin.
• Then on the 1--dimensional line, we must go a distance of \(5/5000 = 0.001\) on the average to capture the 5 nearest neighbors.
• In 2 dimensions, we must go \((0.001)^{1/2}\) to get a square that contains 0.001 of the area.
• In D dimensions, we must go \((0.001)^{1/d}\)
Curse of Dimensionality in k-NN

- With 5000 points in 10 dimensions, we must go 0.501 distance along each attribute in order to find the 5 nearest neighbors.

Curse of Dimensionality

- For a given sample size, there is a maximum number of features above which the performance of our classifier will degrade rather than improve.
- Often the additional information that is lost by discarding some features is (more than) compensated by a more accurate mapping in the lower dimensional space.
Beat the Curse of Dimensionality:
Feature Subset Section

• Feature selection requires
  – A search strategy to select candidate subsets
    • Exhaustive evaluation of subsets is unfeasible
  – An objective function to evaluate “goodness” of these candidates
    • Filters: evaluate feature subsets by their information content, typically interclass distance, information theoretic measures
    • Wrappers: a pattern classifier evaluates feature subsets by their predictive accuracy (resampling or cross-validation)

Search Strategy

• Sequential
  – Naïve sequential feature selection
    • Evaluate each individual feature separately and select M features with highest scores
    • Often work poorly as it doesn’t account for feature dependence
  – Sequential forward selection
  – Sequential backward selection
Sequential Forward Selection

• Simplest greedy search algorithm
  – Starting from empty set, sequentially add feature $x^*$ that results in the highest objective function when combined with features $Y_k$ that have already been selected

1. Start with the empty set $Y_0 = \emptyset$
2. Select the next best feature $x^* = \text{argmax}_{x \in Y_k} [J(Y_k + x)]$
3. Update $Y_{k+1} = Y_k + x^*$; $k = k + 1$
4. Go to 2

• Performs well when the optimal subset has a small number of features
• Unable to remove features that become obsolete after the addition of other features

Sequential Backward Selection

• Works in the opposite direction of SFS
  – Start from full set, sequentially remove the feature $x^-$ that results in the smallest decrease in the objection function

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \text{argmax}_{x \in Y_k} [J(Y_k - x)]$
3. Update $Y_{k+1} = Y_k - x^-$; $k = k + 1$
4. Go to 2

• Works well when the optimal feature subset has a large number of features since SBS spends most of time visiting large subsets
• Main limitation of SBS is its inability to reevaluate the usefulness of a feature after it has been discarded
Filter vs. Wrapper

• Filters
  – Fast execution
  – Generality: evaluates the intrinsic properties of the data, rather than their interaction with a particular classifier
  – Tendency to select large subsets

• Wrappers
  – Accuracy: it's tuned to the specific interactions between the classifier and the data set
  – Ability to generalize: use cross-validation to avoid overfitting
  – Slow execution: classifier training
  – Lack of generality: it’s tied to the bias of the classifier used in the evaluation function. Optimality of the subsets is specific to the classifier.

Beat the Curse of Dimensionality:
Dimensionality Reduction

• Feature combination

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow f\left( \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \right) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = y \quad \text{with } k < d \]

• Ways to do this:
  – Principle Component Analysis (PCA)
  – Fisher Linear Discriminant