Instance Based Learning

Readings: CIML, Chapter 3;
Mitchell, Chapter 8;
Murphy, Chapter 1
Slides arranged by Vibhav Gogate, UT Dallas
sources: course slides are based on material from a variety of
sources, including Tom Dietterich, Carlos Guestrin, Ray Mooney,
Andrew Moore, Andrew Ng and others.

Some Vocabulary

- **Parametric vs. Non-parametric:**
  - **Parametric:**
    - A particular functional form is assumed (e.g., perceptron – next week!)
    - Advantage of simplicity – easy to estimate and interpret
    - May have high bias because the real data may not obey the assumed functional form.
  - **Non-parametric:**
    - Distribution or density estimate is data-driven and relatively few assumptions are made a priori about the functional form.
    - Data determines the model complexity
  - Other terms: Instance-based, Memory-based, Lazy, Case-based, kernel methods...
  - Q: Is a decision tree parametric or non-parametric?

Nearest Neighbor Algorithm

- **Learning Algorithm:**
  - Store training examples
- **Prediction Algorithm:**
  - To classify a new example \(\mathbf{x}\) by finding the training example \((\mathbf{x}', y')\) that is nearest to \(\mathbf{x}\)
  - Guess the class \(y = y'\)

K-Nearest Neighbor Methods

- To classify a new input vector \(\mathbf{x}\), examine the \(k\)-closest training data points to \(\mathbf{x}\) and assign the object to the most frequently occurring class
- Optionally, give closer points larger weights and more distant points smaller weights.

Decision Boundaries

- The nearest neighbor algorithm does not explicitly compute decision boundaries. However, the decision boundaries form a subset of the Voronoi diagram for the training data.

Nearest Neighbor

**When to Consider**
- Instances map to points in \(\mathbb{R}^n\)
- Less than 20 attributes per instance
- Lots of training data

**Advantages**
- Training is very fast
- Learn complex target functions
- Do not lose information

**Disadvantages**
- Slow at query time
- Easily fooled by irrelevant attributes
**Issues**

- Distance measure
  - Most common: Euclidean
- Choosing k
  - Increasing k reduces variance, increases bias
- For high-dimensional space, problem that the nearest neighbor may not be very close at all!
- Memory-based technique. Must make a pass through the data for each classification. This can be prohibitive for large data sets.

**Distance**

- Notation: object with p measurements
  \[ x^i = (x^i_1, x^i_2, \ldots, x^i_p) \]
- Most common distance metric is Euclidean distance:
  \[ d_k(x^i, x^j) = \left( \sum_{k=1}^{p} (x^i_k - x^j_k)^2 \right)^{1/2} \]
- Efficiency trick: using squared Euclidean distance gives same answer, avoids computing square root
- ED makes sense when different measurements are commensurate; each is variable measured in the same units.
- If the measurements are different, say length and weight, it is not clear.

**Standardization**

When variables are not commensurate, we can standardize them by dividing by the sample standard deviation. This makes them all equally important.

The estimate for the standard deviation of \( x_k \):

\[ s_k = \left( \frac{1}{n} \sum_{i=1}^{n} (x^i_k - \bar{x}_k)^2 \right)^{1/2} \]

where \( \bar{x}_k \) is the sample mean:

\[ \bar{x}_k = \frac{1}{n} \sum_{i=1}^{n} x^i_k \]

**Weighted Euclidean distance**

Finally, if we have some idea of the relative importance of each variable, we can weight them:

\[ d_{WE}(i, j) = \left( \sum_{k=1}^{p} w_k (x^i_k - x^j_k)^2 \right)^{1/2} \]

One option: weight each feature by its mutual information with the class.

**Other Distance Metrics**

- Minkowski or L\(_q\) metric:
  \[ d(i, j) = \left( \sum_{k=1}^{p} (x_k(i) - x_k(j))^q \right)^{1/q} \]
- Manhattan, city block or L\(_1\) metric:
  \[ d(i, j) = \sum_{k=1}^{p} |x_k(i) - x_k(j)| \]
- L\(_\infty\)
  \[ d(i, j) = \max_k |x_k(i) - x_k(j)| \]

**The Curse of Dimensionality**

- Nearest neighbor breaks down in high-dimensional spaces because the “neighborhood” becomes very large.
- Suppose we have 5000 points uniformly distributed in the unit hypercube and we want to apply the 5-nearest neighbor algorithm.
- Suppose our query point is at the origin.
  - 1D
    - On a one-dimensional line, we must go a distance of 5/5000 = 0.001 on average to capture the 5 nearest neighbors
  - 2D
    - In two dimensions, we must go sqrt(0.001) to get a square that contains 0.001 of the volume
  - D
    - In d dimensions, we must go (0.001)^{1/2}
Curse of Dimensionality cont.

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

K-NN and irrelevant features

Key Idea: Adjust distance so that things in the same class are close to each other.

Key idea: Formulate this as an optimization problem.

- and rewrite it several times until you have a convex formulation (as a semi-definite program):

Minimize \((1 - \mu) \sum_{j=1}^{K} (y_i - y_j^T) M (x_i - x_j) + \mu \sum_{j=1}^{K} (1 - y_i^T) \varepsilon_j \) subject to:

1. \((y_i - y_j^T) M (x_i - x_j) - (y_i - y_j^T) M (x_i - x_j) \geq 1 - \varepsilon_i^T \)
2. \(\varepsilon_i^T \geq 0\)
3. \(M \geq 0\),

\(M\) represents the distance metric.
(The identity matrix would be standard Euclidean distance.)

The squiggles are slack variables, which are helpers for defining the optimization problem.

The mu balances the two goals — attracting good neighbors and repelling bad neighbors.
Figure 3: Training and test results on the five largest data sets, preprocessed in different ways, and using different variants of kNN classification. We compared principal component analysis (pca), linear discriminant analysis (lda), relevant component analysis (rca), large margin nearest neighbor classification (lmnn), lmnn with multiple passes (mp-lmnn), lmnn with multiple metrics (mm-lmnn), multi-class support vector machines (svm), lmnn classification with the energy based decision rule (lmnn (energy)). All variations of lmnn, rca and lda were applied after pre-processing with pca for general noise reduction. See text and Table 1 for details. The lmnn results consistently outperform pca and lda. The multiple metrics version of lmnn (mm-lmnn) is comparable with multiclass svm on most data sets (with 20news and yaleFaces as only exceptions).

Experimental results were obtained by averaging over multiple runs on randomly generated 70/30 splits of each data set. This procedure was followed with two exceptions: no averaging was done for the Isolet and MNIST data sets, which have pre-defined training/test splits. For all experiments reported in this paper, the number of target neighbors \( k \) was set to \( k = 3 \), and the weighting parameter \( \mu \) in Eqs. (14-15) was set to \( \mu = 0.5 \). Though we experimented with different settings, the results from LMNN classification appeared fairly insensitive to the values of these parameters.

The main results on the five largest data sets are shown in Fig. 3. (See Table 1 for a complete listing of results, including those for various extensions of LMNN classification described in section 5.) All training error rates reported are leave-one-out estimates. To break ties among different...

Nearest Neighbor Summary

- **Advantages**
  - Variable-sized hypothesis space
  - Learning is extremely efficient
  - However, growing a good kd-tree can be expensive
  - Very flexible decision boundaries
  - Easy to implement
  - Can be very effective

- **Disadvantages**
  - Distance function must be carefully chosen
  - Irrelevant or correlated features must be eliminated
  - Typically cannot handle more than 30 features
  - Computational costs: Memory and classification-time computation