Nearest neighbor classifiers

COMS 4721
Example: OCR for digits

- Classify images of handwritten digits by the (actual) digits they depict.
- Classification problem: $\mathcal{Y} = \text{discrete set}$
Nearest neighbor (NN) classifier

• **Given**: labeled examples $D := \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$

• **Predictor** $\hat{f}_D: \mathcal{X} \to \mathcal{Y}$:

  On input $x \in \mathcal{X}$:
  1. Find the point $x_i$ among $\{x_i\}_{i=1}^n$ "closest" to $x$ (nearest neighbor)
  2. Return $y_i$
How to measure distance?

• For points in $\mathbb{R}^d$, a default choice for distance is the *Euclidean distance* (also called $\ell_2$ distance).

$$
\|u - v\|_2 = \sqrt{\sum_{j=1}^{d} (u_j - v_j)^2}
$$

Grayscale 28×28 pixel images. Treat as *vectors* (of 784 features) that live in $\mathbb{R}^{784}$. 
Example: OCR for digits with NN classifier

• Classify images of handwritten digits by the digits they depict.

\[
\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
\]

• \( \mathcal{X} = \mathbb{R}^{768}, \mathcal{Y} = \{0,1,2,3,4,5,6,7,8,9\} \)

• **Given:** labeled examples \( D := \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y} \).

• Construct NN classifier \( \hat{f}_D \) using \( D \).

• **Question:** How good is this classifier?
Error rate

• **Error rate** of classifier $f$ on a set of labeled examples $D$:

$$\text{err}_D(f) := \frac{|\{(x, y) \in D : f(x) \neq y\}|}{|D|}$$

(on what fraction of $D$ does $f$ disagree with the paired label?)

• (Sometimes, we’ll write this as $\text{err}(f, D)$.)

• **Question**: What is $\text{err}_D(\hat{f}_D)$?
A better way to evaluate the classifier

• Split the labeled examples \( \{(x_i, y_i)\}_{i=1}^{n} \) into two sets
  • Training data (\( S \))
  • Test data (\( T \))

• Only use \textit{training data} to construct the NN classifier \( \hat{f}_S \).
  • Training error rate of \( \hat{f}_S \): \( \text{err}_S(\hat{f}_S) = 0\% \)

• Use \textit{test data} to evaluate accuracy of \( \hat{f}_S \).
  • Test error rate of \( \hat{f}_S \): \( \text{err}_T(\hat{f}_S) = 3.09\% \)
Diagnostics

• Some examples of NN classifier mistakes (test point in $T$, nearest neighbor in $S$)

  28  35  54

• First mistake (correct label is “2”) could’ve been avoided by looking at the three nearest neighbors (whose labels are “8”, “2”, and “2”):

  2  8  2  2

  Test point  Three nearest neighbors
**$k$-nearest neighbors ($k$-NN) classifier**

- **Given:** labeled examples $D := \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$

- **Predictor** $\hat{f}_{D,k} : \mathcal{X} \rightarrow \mathcal{Y}$:

  On input $x \in \mathcal{X}$:
  1. Find the $k$ points $x_{i_1}, x_{i_2}, ..., x_{i_k}$ among $\{x_i\}_{i=1}^n$ “closest” to $x$ (the $k$ nearest neighbors)
  2. Return plurality of $y_{i_1}, y_{i_2}, ..., y_{i_k}$

(Break ties arbitrarily in both steps.)
Effect of $k$

- Smaller $k$: smaller training error.
- Larger $k$: higher training error, but predictions are more “stable” due to voting.

OCR digits classification:

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test error rate</td>
<td>3.09%</td>
<td>2.95%</td>
<td>3.12%</td>
<td>3.06%</td>
<td>3.41%</td>
</tr>
</tbody>
</table>
Picking $k$

• **Simplest approach: use a hold-out set:**
  1. Pick a subset $V \subset S$ (*hold-out set, or validation set*).
  2. For each $k \in \{1, 3, 5, \ldots\}$:
     - Construct $k$-NN classifier $\hat{f}_{S \setminus V, k}$ using $S \setminus V$
     - Compute error rate of $\hat{f}_{S \setminus V, k}$ on $V$ ("hold-out error rate")
  3. Pick the $k$ that gives the smallest hold-out error rate.
Some better distances

- **Strings: edit distance**
  - $d(u, v) = \# \text{ insertions/deletions/mutations needed to change } u \text{ to } v$

- **Images: shape context distance**
  - $d(u, v) = \text{ how much "warping" is required to change } u \text{ to } v$

OCR digits classification:

<table>
<thead>
<tr>
<th>Distance</th>
<th>$\ell_2$</th>
<th>$\ell_3$</th>
<th>Tangent</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test error rate</td>
<td>3.09%</td>
<td>2.83%</td>
<td>1.10%</td>
<td>0.63%</td>
</tr>
</tbody>
</table>
Noisy features

**Caution:** nearest neighbors can be broken by noisy features!
Important questions

1. How good is the classifier learned using NN on your problem?
2. Is NN a good learning algorithm, in general?
Statistical learning theory

• **Basic assumption:** labeled examples \( \{(x_i, y_i)\}_{i=1}^{n} \) come from same source as future unlabeled examples

Collection of labeled examples from \( P \) → **Learning algorithm** → Learned predictor

New unlabeled example from \( P \) → Prediction

**Assumption:** \( \{(x_i, y_i)\}_{i=1}^{n} \) is an *i.i.d. sample* from a probability distribution \( P \) over \( \mathcal{X} \times \mathcal{Y} \)
Prediction error rate

• Define the (true) error rate of a classifier \( f : \mathcal{X} \rightarrow \mathcal{Y} \) w.r.t. \( P \) to be
  \[
  \text{err}_P(f) := P(f(X) \neq Y)
  \]
  where \((X, Y) \sim P\).

• Let \( \hat{f}_S \) be a classifier trained using labeled examples \( S \).
• The true error rate of \( \hat{f}_S \) is
  \[
  \text{err}_P(\hat{f}_S) = P(\hat{f}_S(X) \neq Y).
  \]
• We cannot compute this without knowing \( P \).
Estimating the true error rate

• Suppose we split \( \{(x_i, y_i)\}_{i=1}^n \) into \( S \) and \( T \), and \( \hat{f}_S \) is only based on \( S \).

• Under the i.i.d. sample assumption, \( \hat{f}_S \) and \( T \) are independent, and test error rate of \( \hat{f}_S \) is an unbiased estimate of true error rate of \( \hat{f}_S \).

• If \( |T| = m \), then the test error rate of \( \hat{f}_S \) is a binomial random variable (scaled by \( 1/m \)):

\[
m \cdot \text{err}_T(\hat{f}_S) \sim \text{Bin} \left( m, \text{err}_P(\hat{f}_S) \right)
\]

• The expected value of \( \text{err}_T(\hat{f}_S) \) is \( \text{err}_P(\hat{f}_S) \). (Unbiased estimator)

• Its standard deviation is at most \( 1/\sqrt{m} \).
Limits of prediction

• Binary classification: \( Y = \{0, 1\} \)

• Think of \( P \) as being comprised of two parts:
  • Marginal distribution of \( X \) (i.e., a distribution over \( \mathcal{X} \)): call this \( \mu \).
  • Conditional distribution of \( Y \) given \( X = x \), for each \( x \in \mathcal{X} \):
    \[ \eta(x) := P(Y = 1|X = x) \]

• If \( \eta(x) \) is 0 or 1 for all \( x \in \mathcal{X} \) where \( \mu(x) > 0 \), then the optimal error rate is zero (i.e., \( \min_{f} \text{err}_{P}(f) = 0 \)).

• Otherwise it is non-zero.
Bayes optimality

• What is the classifier with smallest true error rate?

\[ f^*(x) := \begin{cases} 
0, & \eta(x) \leq 1/2 \\
1, & \eta(x) > 1/2 
\end{cases} \]

• \( f^* \) is called the Bayes (optimal) classifier

\[ \text{err}_p(f^*) = \min_{f} \text{err}_p(f) = \mathbb{E}[\min\{\eta(X), 1 - \eta(X)\}] \]

• Its error rate is called the Bayes (optimal) error.
Consistency

• We say a learning algorithm $A$ is consistent if
  \[
  \lim_{n \to \infty} \mathbb{E}[\text{err}_P(\hat{f}_n)] = \min_f \text{err}_P(f)
  \]
  where $\hat{f}_n$ denotes classifier learned by $A$ from i.i.d. sample of size $n$.

• **Theorem** [e.g., Cover and Hart, 1967]. Assume $\eta$ is continuous. Then:
  • 1-NN is consistent if $\min_f \text{err}_P(f) = 0$.
  • $k$-NN is consistent provided that $k := k_n$ is chosen as an increasing but sublinear function of $n$:
    \[
    \lim_{n \to \infty} k_n = \infty, \quad \lim_{n \to \infty} \frac{k_n}{n} = 0.
    \]