Supervised Learning: Nearest Neighbors

Prof. Adriana Kovashka
University of Pittsburgh
February 1, 2016
Today: Supervised Learning Part I

• Basic formulation of the simplest classifier: K Nearest Neighbors

• Example uses

• Generalizing the distance metric and weighting neighbors differently

• Problems:
  – The curse of dimensionality
  – Picking K
  – Approximation strategies
Key Idea

• Supervised learning: We want to learn to predict, for a new data point $x$, its label $y$ (e.g. spam / not spam)

• Don’t learn an explicit function $F: X \rightarrow Y$

• Keep all training data $\{X, Y\}$

• For a test example $x$, find the training example $x_i$ closest to it (e.g. using Euclidean distance)

• Then copy the target label $y_i$ as the label for $x$
Related Methods

• Instance-based methods
• Exemplar methods
• Memory-based methods
• Non-parametric methods
**Nearest Neighbor Classifier**

\[ f(x) = \text{label of the training example nearest to } x \]

- All we need is a distance function for our inputs
- No training required!
**K-Nearest Neighbors Classifier**

- For a new point, find the $k$ closest points from training data (e.g. $k=5$)
- Labels of the $k$ points “vote” to classify

If query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

Slide credit: David Lowe
1-nearest neighbor
3-nearest neighbor
What are the tradeoffs of having a too large $k$? Too small $k$?
Instance/Memory-based Learning

Four things make a memory based learner:

• A distance metric

• How many nearby neighbors to look at?

• A weighting function (optional)

• How to fit with the local points?
1-Nearest Neighbor

Four things make a memory based learner:

• A distance metric
  – Euclidean (and others)

• How many nearby neighbors to look at?
  – 1

• A weighting function (optional)
  – Not used

• How to fit with the local points?
  – Just predict the same output as the nearest neighbor

Slide credit: Carlos Guestrin
k-Nearest Neighbor

Four things make a memory based learner:

- A *distance metric*  
  - Euclidean (and others)

- How many nearby neighbors to look at?  
  - k

- A *weighting function (optional)*  
  - Not used

- How to fit with the local points?  
  - Just predict the average output among the nearest neighbors
Formal Definition

• Classification:

• Regression:
Example: Predict where this picture was taken
Example: Predict where this picture was taken
Example: Predict where this picture was taken
6+ million geotagged photos by 109,788 photographers

Scene Matches

Scene Matches

Scene Matches

The Importance of Data

Collaborative Filtering (Netflix)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From Y. Koren of BellKor team

4 + 9 = 13
1 + 1 + 1 = 3
1 + 4 = 5
1 + 9 + 4 = 14
0 + 0 + 0 = 0

Slide credit: Alexander Ihler
Text representations

- "Bag of words"
  - Remember word counts but not order

- Example:

Rain and chilly weather didn't keep thousands of paradegoers from camping out Friday night for curbside seats for today's parade.

"I want to party all night," said Tyne Gaudo of Glendale, who spent the last night of the year camping out on Boulevard with a group of friends.

Whether they came for the partying or the chance in for a long night. Rain continued into the night, temperatures were expected to dip down into the.
Latent Semantic Indexing (LSI)

- PCA for text data
- Create a giant matrix of words in docs
  - “Word j appears” = feature $x_{ij}$
  - “in document i” = data example $i$

- PCA on this matrix provides a new representation
  - Document comparison
  - Fuzzy search (“concept” instead of “word” matching)
Latent Semantic Indexing (LSI) worked-out example:

Singular Value Decomposition

- Alternative method to calculate (still subtract mean 1st)
- Decompose $X = U S V^T$
  - Orthogonal: $X^T X = V S S V^T = V D V^T$
  - $X X^T = U S S U^T = U D U^T$

- $U*S$ matrix provides coefficients
  - Example $x_i = U_{i,1} S_{11} v_1 + U_{i,2} S_{22} v_2 + \ldots$

- Gives the least-squares approximation to $X$ of this form

$$X_{N \times D} \approx U_{N \times K} S_{K \times K} V_{K \times D}^T$$
k-Nearest Neighbor

Four things make a memory based learner:

• **A distance metric**
  – Euclidean (and others)

• **How many nearby neighbors to look at?**
  – k

• **A weighting function (optional)**
  – Not used

• **How to fit with the local points?**
  – Just predict the average output among the nearest neighbors

Slide credit: Carlos Guestrin
Distances

• Suppose I want to charge my overall distance more for differences in $x_2$ direction as opposed to $x_1$ direction
• Setup A: equal weighing on all directions
• Setup B: more weight on $x_2$ direction
• Will my neighborhoods be longer in the $x_1$ or $x_2$ direction?
Voronoï partitioning

- Nearest neighbor regions
- All points in a region are closer to the seed in that region than to any other seed (black dots = seeds)
Multivariate distance metrics

Suppose the input vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are two dimensional:

$\mathbf{x}_1 = (x_{11}, x_{12}), \mathbf{x}_2 = (x_{21}, x_{22}), \ldots, \mathbf{x}_N = (x_{N1}, x_{N2})$.

Dist\(\mathbf{x}_i, \mathbf{x}_j\) = \((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2\)

Dist\(\mathbf{x}_i, \mathbf{x}_j\) = \((x_{i1} - x_{j1})^2 + (3x_{i2} - 3x_{j2})^2\)

The relative scalings in the distance metric affect region shapes

Slide credit: Carlos Guestrin
Distance metrics

• Euclidean:

• Mahalanobis:

• Minkowski:
Distance metrics

Voronoi diagrams of 20 points under two different metrics

Euclidean distance

Manhattan distance

Figures from Wikipedia
Another generalization: Weighted K-NNs

• Neighbors weighted differently:

• Extremes
  – Bandwidth = infinity: prediction is dataset average
  – Bandwidth = zero: prediction becomes 1-NN
Kernel Regression/Classification

Four things make a memory based learner:

• A distance metric
  – Euclidean (and others)

• How many nearby neighbors to look at?
  – All of them

• A weighting function (optional)
  – \( w_i = \exp(-d(x_i, \text{query})^2 / \sigma^2) \)
  – Nearby points to the query are weighted strongly, far points weakly. The \( \sigma \) parameter is the **Kernel Width**. Very important.

• How to fit with the local points?
  – (Regression) Predict the weighted average of the outputs, i.e. \( \Sigma w_i y_i / \Sigma w_i \)
Problems with Instance-Based Learning

• Too many features?
  – Doesn’t work well if large number of irrelevant features, distances overwhelmed by noisy features
  – Distances become meaningless in high dimensions (the curse of dimensionality)

• What is the impact of the value of K?

• Expensive
  – No learning: most real work done during testing
  – For every test sample, must search through all dataset – very slow!
  – Must use tricks like approximate nearest neighbor search
  – Need to store all training data

Adapted from Dhruv Batra
Curse of Dimensionality

How many neighborhoods are there?

Fruit data

#bins = 10x10
\[ d = 2 \]

#bins = \(10^d\)
\[ d = 1000 \]

Atoms in the universe
\[ \sim 10^{80} \]

Subhransu Maji (UMASS)
Curse of Dimensionality

- Consider: Sphere of radius 1 in d-dims

- Consider: an outer $\varepsilon$-shell in this sphere

- What is $\frac{\text{shell volume}}{\text{sphere volume}}$?
Curse of Dimensionality

Figure 1.22 from Bishop
Curse of Dimensionality

• Problem: In very high dimensions, all points are equally close

• This problem applies to all types of classifiers, not just K-NN
Problems with Instance-Based Learning

• Too many features?
  – Doesn’t work well if large number of irrelevant features, distances overwhelmed by noisy features
  – Distances become meaningless in high dimensions (the curse of dimensionality)

• What is the impact of the value of $K$?

• Expensive
  – No learning: most real work done during testing
  – For every test sample, must search through all dataset – very slow!
  – **Must use tricks like approximate nearest neighbor search**
  – Need to store all training data

Adapted from Dhruv Batra
kNN Decision Boundary

- Increasing $k$ \underline{complicates} decision boundary
**kNN Decision Boundary**

- Increasing $k$ “simplifies” decision boundary
  - Majority voting means less emphasis on individual points

$k = 1$

$k = 3$
kNN Decision Boundary

- Increasing $k$ “simplifies” decision boundary
  - Majority voting means less emphasis on individual points

$K = 5$

$K = 7$
kNN Decision Boundary

- Increasing k “simplifies” decision boundary
  - Majority voting means less emphasis on individual points

\[ K = 25 \]
Error rates and K

Use a validation set to pick K

Error on Test Data

Error on Training Data

K (# neighbors)

Predictive Error

K=1? Zero error! Training data have been memorized...

Best value of K

Too complex

Slide credit: Alexander Ihler
Problems with Instance-Based Learning

• Too many features?
  – Doesn’t work well if large number of irrelevant features, distances overwhelmed by noisy features
  – Distances become meaningless in high dimensions (the curse of dimensionality)

• What is the impact of the value of $K$?

• Expensive
  – No learning: most real work done during testing
  – For every test sample, must search through all dataset – very slow!
  – Must use tricks like approximate nearest neighbor search
  – Need to store all training data

Adapted from Dhruv Batra
Approximate distance methods

• Build a balanced tree of data points (*kd*-trees), splitting along different dimensions
• Go down tree (starting from root) and at each point compute “current best” guess of nearest neighbor
• Intelligently eliminate parts of the search space if they cannot contain a better “current best”
• Only search for neighbors up until some budget exhausted
Approximate kNN

- k-d tree: $O(\log N)$ query time

Approximate kNN

- k-d tree: $O(\log N)$ query time

![Fruit data graph with points and median split]

Approximate kNN

- k-d tree: $O(\log N)$ query time

Fruit data

split at the median

Approximate kNN

- k-d tree: $O(\log N)$ query time

Fruit data

split at the median

height (cm)

width (cm)

Approximate kNN

- k-d tree: $O(\log N)$ query time

Approximate kNN

- k-d tree: $O(\log N)$ query time

(split at the median)

Figure from Szeliski
Summary

• K-Nearest Neighbor is the most basic and simplest to implement classifier
• Cheap at training time, expensive at test time
• Unlike other methods we’ll see later, naturally works for any number of classes
• Pick K through a validation set, use approximate methods for finding neighbors
• Success of classification depends on the amount of data and the meaningfulness of the distance function