Fundamentals of Machine Learning for Predictive Data Analytics

Chapter 5: Similarity-based Learning

Sections 5.1, 5.2, 5.3

John Kelleher and Brian Mac Namee and Aoife D'Arcy

john.d.kelleher@dit.ie brian.macnamee@ucd.ie aoife@theanalyticsstore.com

1. **Big Idea**
2. **Fundamentals**
   - Feature Space
   - Distance Metrics
3. **Standard Approach: The Nearest Neighbor Algorithm**
   - A Worked Example
4. **Epilogue**
5. **Summary**

Big Idea

- The year is 1798 and you are Lieutenant-Colonel David Collins of HMS Calcutta who is exploring the region around Hawkesbury River, in New South Wales.
- After an expedition up the river one of the men tells you that he saw a strange animal near the river.
- You ask him to describe the animal to you and he explains that he didn't see it very well but that he did notice that it had webbed feet and a duck-bill snout, and that it growled at him.
- In order to plan the expedition for the next day you decide that you need to classify the animal so that you can figure out whether it is dangerous to approach it or not.

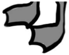
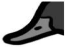
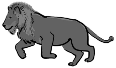
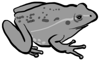
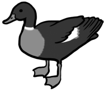
	Grrrh!			**Score**
	✓	✗	✗	1
	✗	✓	✗	1
	✗	✓	✓	2

Figure: Matching animals you remember to the features of the unknown animal described by the sailor. Note: The images used in this figure were created by Jan Gillbank for the English for the Australian Curriculum website (http://www.e4ac.edu.au) and are used under the Create Commons Attribution 3.0 Unported licence (http://creativecommons.org/licenses/by/3.0). The images were sourced via Wikimedia Commons.

- The process of classifying an unknown animal by matching the features of the animal against the features of animals you can remember neatly encapsulates the big idea underpinning similarity-based learning:

> *if you are trying to classify something then you should search your memory to find things that are similar and label it with the same class as the most similar thing in your memory*

- One of the simplest and best known machine learning algorithms for this type of reasoning is called the nearest neighbor algorithm.

Fundamentals

- The fundamentals of similarity-based learning are:
    - Feature space
    - Similarity metrics

Feature Space

Table: The speed and agility ratings for 20 college athletes labelled with the decisions for whether they were drafted or not.

ID	Speed	Agility	Draft
1	2.50	6.00	No
2	3.75	8.00	No
3	2.25	5.50	No
4	3.25	8.25	No
5	2.75	7.50	No
6	4.50	5.00	No
7	3.50	5.25	No
8	3.00	3.25	No
9	4.00	4.00	No
10	4.25	3.75	No
11	2.00	2.00	No
12	5.00	2.50	No
13	8.25	8.50	No
14	5.75	8.75	Yes
15	4.75	6.25	Yes
16	5.50	6.75	Yes
17	5.25	9.50	Yes
18	7.00	4.25	Yes
19	7.50	8.00	Yes
20	7.25	5.75	Yes

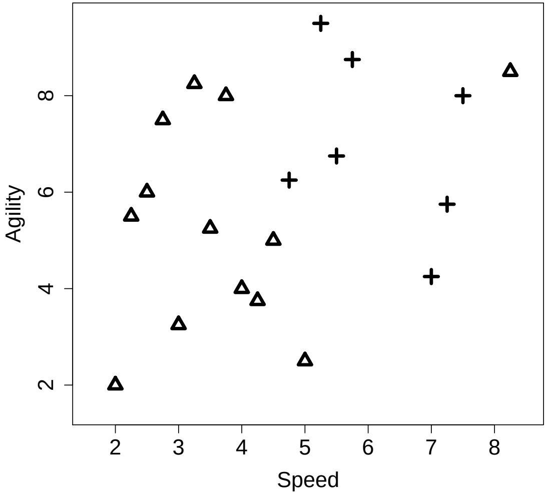


Figure: A feature space plot of the data in Table 2 [25]. The triangles represent *'Non-draft'* instances and the crosses represent the *'Draft'* instances.

Feature Space

- A feature space is an abstract n-dimensional space that is created by taking each of the descriptive features in an ABT to be the axes of a reference space and each instance in the dataset is mapped to a point in the feature space based on the values of its descriptive features.

- A similarity metric measures the similarity between two instances according to a feature space
- Mathematically, a metric must conform to the following four criteria:
  1. Non-negativity: $\textit{metric}(\mathbf{a}, \mathbf{b}) \geq 0$
  2. Identity: $\textit{metric}(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b}$
  3. Symmetry: $\textit{metric}(\mathbf{a}, \mathbf{b}) = \textit{metric}(\mathbf{b}, \mathbf{a})$
  4. Triangular Inequality: $\textit{metric}(\mathbf{a}, \mathbf{b}) \leq \textit{metric}(\mathbf{a}, \mathbf{c}) + \textit{metric}(\mathbf{b}, \mathbf{c})$

  where $\textit{metric}(\mathbf{a}, \mathbf{b})$ is a function that returns the distance between two instances **a** and **b**.

- One of the best known metrics is Euclidean distance which computes the length of the straight line between two points. Euclidean distance between two instances **a** and **b** in a *m*-dimensional feature space is defined as:

$$Euclidean(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{m} (\mathbf{a}[i] - \mathbf{b}[i])^2} \tag{1}$$

Example

The Euclidean distance between instances d_{12} (SPEED$=$ 5.00, AGILITY$=$ 2.5) and d_5 (SPEED$=$ 2.75,AGILITY$=$ 7.5) in Table 2 [25] is:

Example

The Euclidean distance between instances d_{12} (SPEED$=$ 5.00, AGILITY$=$ 2.5) and d_5 (SPEED$=$ 2.75,AGILITY$=$ 7.5) in Table 2 [25] is:

$$\begin{aligned} \mathit{Euclidean}(\langle 5.00, 2.50\rangle, \langle 2.75, 7.50\rangle) &= \sqrt{(5.00-2.75)^2 + (2.50-7.50)^2} \\ &= \sqrt{30.0625} = 5.4829 \end{aligned}$$

- Another, less well known, distance measure is the Manhattan distance or taxi-cab distance.
- The Manhattan distance between two instances **a** and **b** in a feature space with *m* dimensions is:[1]

$$Manhattan(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m} abs(\mathbf{a}[i] - \mathbf{b}[i]) \tag{2}$$

[1] The *abs*() function surrounding the subtraction term indicates that we use the absolute value, i.e. non-negative value, when we are summing the differences; this makes sense because distances can't be negative.

Distance Metrics

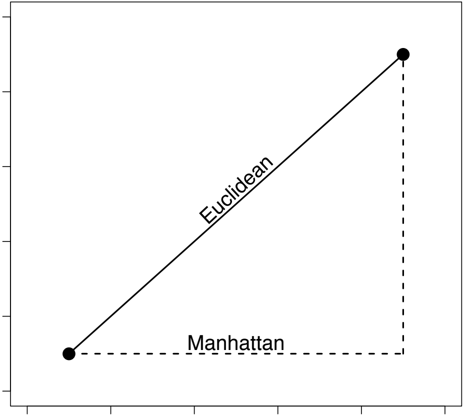


Figure: The Manhattan and Euclidean distances between two points.

Example

The Manhattan distance between instances d_{12} (SPEED$=5.00$, AGILITY$=2.5$) and d_5 (SPEED$=2.75$,AGILITY$=7.5$) in Table 2 [25] is:

Example

The Manhattan distance between instances d_{12} (SPEED$=$ 5.00, AGILITY$=$ 2.5) and d_5 (SPEED$=$ 2.75,AGILITY$=$ 7.5) in Table 2 [25] is:

$$\begin{aligned} \mathit{Manhattan}(\langle 5.00, 2.50\rangle, \langle 2.75, 7.50\rangle) &= \mathit{abs}(5.00 - 2.75) + \mathit{abs}(2.5 - 7.5) \\ &= 2.25 + 5 = 7.25 \end{aligned}$$

- The Euclidean and Manhattan distances are special cases of Minkowski distance
- The Minkowski distance between two instances **a** and **b** in a feature space with m descriptive features is:

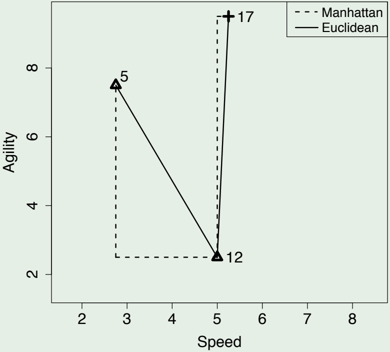
$$Minkowski(\mathbf{a},\mathbf{b}) = \left(\sum_{i=1}^{m} abs(\mathbf{a}[i] - \mathbf{b}[i])^{p}\right)^{\frac{1}{p}} \tag{3}$$

  where different values of the parameter p result in different distance metrics
- The Minkowski distance with $p = 1$ is the Manhattan distance and with $p = 2$ is the Euclidean distance.

- The larger the value of p the more emphasis is placed on the features with large differences in values because these differences are raised to the power of p.

Example

Instance ID	Instance ID	Manhattan (Minkowski p=1)	Euclidean (Minkowski p=2)
12	5	7.25	5.4829
12	17	7.25	8.25



The Manhattan and Euclidean distances between instances $\mathbf{d}_{12}$ (SPEED= 5.00, AGILITY= 2.5) and $\mathbf{d}_5$ (SPEED= 2.75, AGILITY= 7.5) and between instances $\mathbf{d}_{12}$ and $\mathbf{d}_{17}$ (SPEED= 5.25, AGILITY= 9.5).

Standard Approach: The Nearest Neighbor Algorithm

The Nearest Neighbour Algorithm

```
Require: set of training instances
Require: a query to be classified
  Iterate across the instances in memory and find the instance that is shortest distance from the query position in the feature space.
  Make a prediction for the query equal to the value of the target feature of the nearest neighbor.
```

A Worked Example

Table: The speed and agility ratings for 20 college athletes labelled with the decisions for whether they were drafted or not.

ID	Speed	Agility	Draft
1	2.50	6.00	No
2	3.75	8.00	No
3	2.25	5.50	No
4	3.25	8.25	No
5	2.75	7.50	No
6	4.50	5.00	No
7	3.50	5.25	No
8	3.00	3.25	No
9	4.00	4.00	No
10	4.25	3.75	No
11	2.00	2.00	No
12	5.00	2.50	No
13	8.25	8.50	No
14	5.75	8.75	Yes
15	4.75	6.25	Yes
16	5.50	6.75	Yes
17	5.25	9.50	Yes
18	7.00	4.25	Yes
19	7.50	8.00	Yes
20	7.25	5.75	Yes

Example

- Should we draft an athlete with the following profile:

$$\textsc{Speed}= 6.75,\ \textsc{Agility}= 3$$

A Worked Example

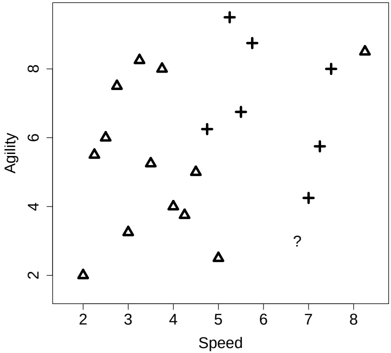


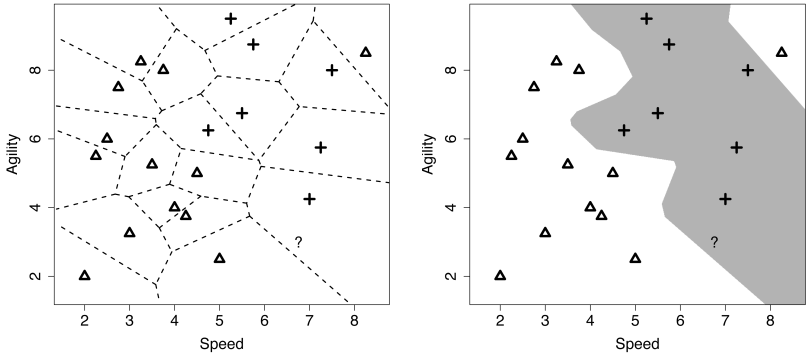
Figure: A feature space plot of the data in Table 2 [25] with the position in the feature space of the query represented by the ? marker. The triangles represent *'Non-draft'* instances and the crosses represent the *'Draft'* instances.

A Worked Example

Table: The distances (Dist.) between the query instance with SPEED = 6.75 and AGILITY = 3.00 and each instance in Table 2 [25].

ID	SPEED	AGILITY	DRAFT	Dist.
18	7.00	4.25	yes	1.27
12	5.00	2.50	no	1.82
10	4.25	3.75	no	2.61
20	7.25	5.75	yes	2.80
9	4.00	4.00	no	2.93
6	4.50	5.00	no	3.01
8	3.00	3.25	no	3.76
15	4.75	6.25	yes	3.82
7	3.50	5.25	no	3.95
16	5.50	6.75	yes	3.95
11	2.00	2.00	no	4.85
19	7.50	8.00	yes	5.06
3	2.25	5.50	no	5.15
1	2.50	6.00	no	5.20
13	8.25	8.50	no	5.70
2	3.75	8.00	no	5.83
14	5.75	8.75	yes	5.84
5	2.75	7.50	no	6.02
4	3.25	8.25	no	6.31
17	5.25	9.50	yes	6.67

A Worked Example



(a) Voronoi tessellation (b) Decision boundary ($k = 1$)

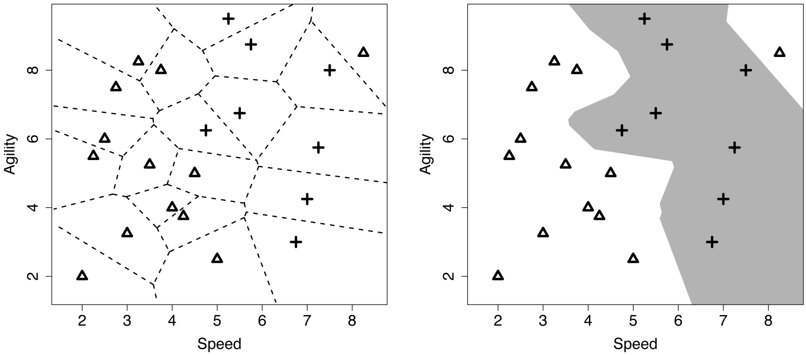
Figure: (a) The Voronoi tessellation of the feature space for the dataset in Table 2 [25] with the position of the query represented by the ? marker; (b) the decision boundary created by aggregating the neighboring Voronoi regions that belong to the same target level.

- One of the great things about nearest neighbour algorithms is that we can add in new data to update the model very easily.

Table: The extended version of the college athletes dataset.

ID	Speed	Agility	Draft
1	2.50	6.00	no
2	3.75	8.00	no
3	2.25	5.50	no
4	3.25	8.25	no
5	2.75	7.50	no
6	4.50	5.00	no
7	3.50	5.25	no
8	3.00	3.25	no
9	4.00	4.00	no
10	4.25	3.75	no
11	2.00	2.00	no
12	5.00	2.50	no
13	8.25	8.50	no
14	5.75	8.75	yes
15	4.75	6.25	yes
16	5.50	6.75	yes
17	5.25	9.50	yes
18	7.00	4.25	yes
19	7.50	8.00	yes
20	7.25	5.75	yes
21	6.75	3.00	yes

A Worked Example



(a) Voronoi tessellation (b) Decision boundary ($k = 1$)

Figure: (a) The Voronoi tessellation of the feature space when the dataset has been updated to include the query instance; (b) the updated decision boundary reflecting the addition of the query instance in the training set.

Epilogue

- Returning to 1798 and HMS Calcutta, the next day you accompany your men on the expedition up the river and you encounter the strange animal the sailor had described to you.
- This time when you see the animal yourself you realize that it definitely isn't a duck!
- It turns out that you and your men are the first Europeans to encounter a platypus[2].

[2] The story recounted here of the discovery of the platypus is loosely based on real events.

Figure: A duck-billed platypus.The platypus image used in here was created by Jan Gillbank for the English for the Australian Curriculum website (`http://www.e4ac.edu.au`) and are used under the Create Commons Attribution 3.0 Unported licence (`http://creativecommons.org/licenses/by/3.0`). The image was sourced via Wikimedia Commons.

- This epilogue illustrates two important, and related, aspects of supervised machine learning:
  1. supervised machine learning is based on the stationarity assumption which states that the data doesn't change - remains stationary - over time.
  2. in the context of classification, supervised machine learning creates models that distinguish between the classes that are present in the dataset they are induced from. So, if a classification model is trained to distinguish between lions, frogs and ducks, the model will classify a query as being either a lion, a frog or a duck; even if the query is actually a platypus.

Summary

- Similarity-based prediction models attempt to mimic a very human way of reasoning by basing predictions for a target feature value on the most similar instances in memory—this makes them easy to interpret and understand.
- This advantage should not be underestimated as being able to understand how the model works gives people more confidence in the model and, hence, in the insight that it provides.

- The inductive bias underpinning similarity-based classification is that things that are similar (i.e., instances that have similar descriptive features) belong to the same class.
- The nearest neighbor algorithm creates an implicit global predictive model by aggregating local models, or neighborhoods.
- The definition of these neighborhoods is based on proximity within the feature space to the labelled training instances.
- Queries are classified using the label of the training instance defining the neighborhood in the feature space that contains the query.

1. **Big Idea**
2. **Fundamentals**
   - Feature Space
   - Distance Metrics
3. **Standard Approach: The Nearest Neighbor Algorithm**
   - A Worked Example
4. **Epilogue**
5. **Summary**