THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

INTRODUCTION TO MACHINE LEARNING

DIT 45100

THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Module 4

Non-Parametric & Probabilistic Algorithms

THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Nearest Neighbors Classifier

KNN

Big Idea

- Feature Similarity

Big Idea

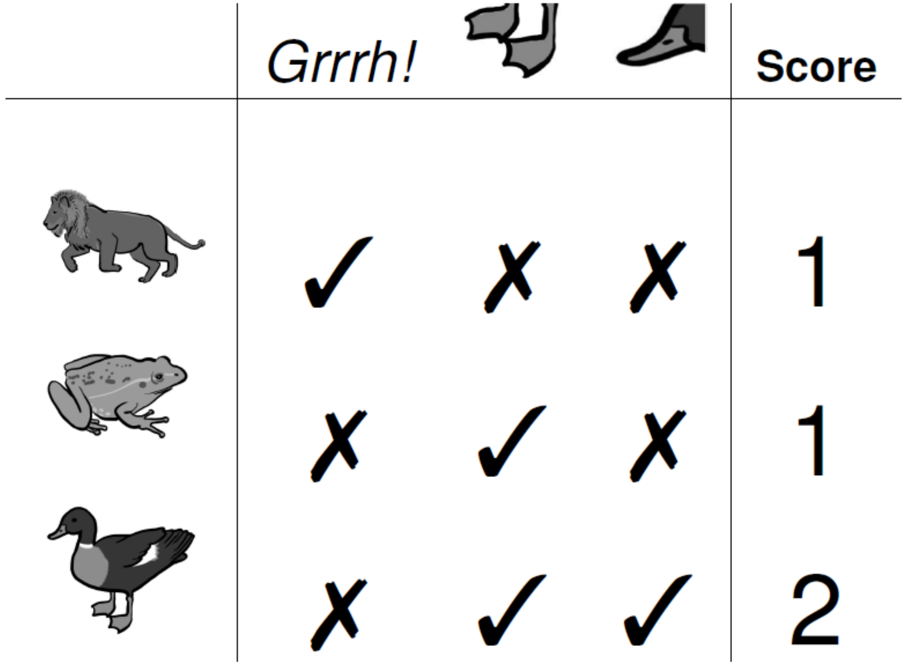


Figure: Matching animals you remember to the features of the unknown animal described by the sailor.

Big Idea

- 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.

THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Big Idea

Figure: A duck-billed platypus.

Big Idea

- This epilogue illustrates two important, and related, aspects of supervised machine learning:
  - Supervised machine learning is based on the stationarity assumption which states that the data doesn’t change - remains stationary - over time.
  - 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.

THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Fundamentals

- Feature Space
- Similarity Metrics

Feature Space

- A feature space is an abstract n-dimensional space that is created by taking each of the descriptive features in a dataset 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 scatter plot is a visual representation of data in a feature space

THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

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

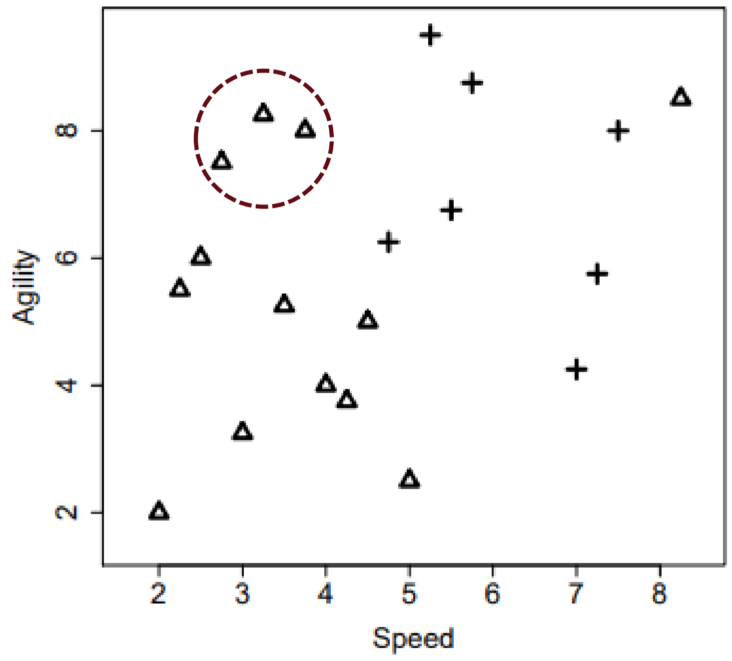


Figure: A feature space plot of the data in Table 2 [25]. The triangles represent *'Non-draft'* instances and the crosses represent the *'Draft'* instances.

Measures of Similarity

- 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: $metric(\mathbf{a}, \mathbf{b}) \geq 0$
  2. Identity: $metric(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b}$
  3. Symmetry: $metric(\mathbf{a}, \mathbf{b}) = metric(\mathbf{b}, \mathbf{a})$
  4. Triangular Inequality: $metric(\mathbf{a}, \mathbf{b}) \leq metric(\mathbf{a}, \mathbf{c}) + metric(\mathbf{b}, \mathbf{c})$

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

Measures of Similarity

- 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:

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

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:

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

Measures of Similarity

- 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:

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

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:

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

Measures of Similarity

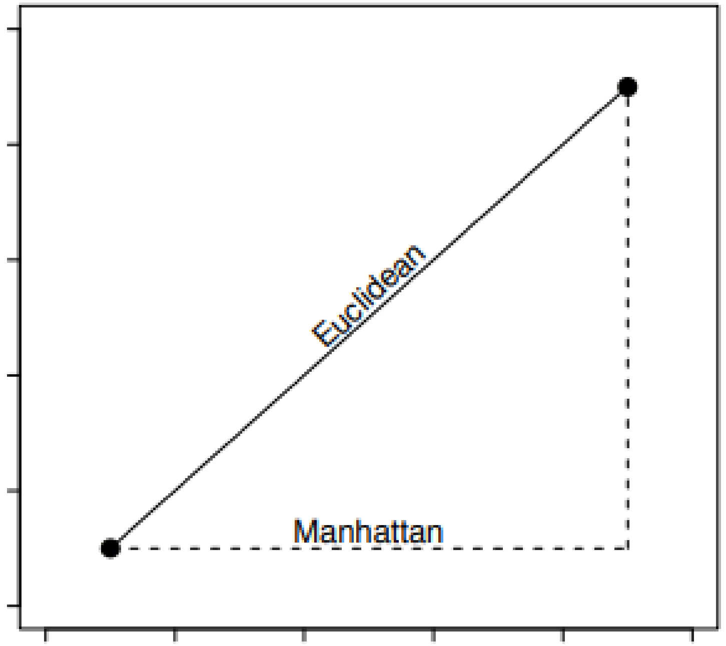


Figure: The Manhattan and Euclidean distances between two points.

Measures of Similarity

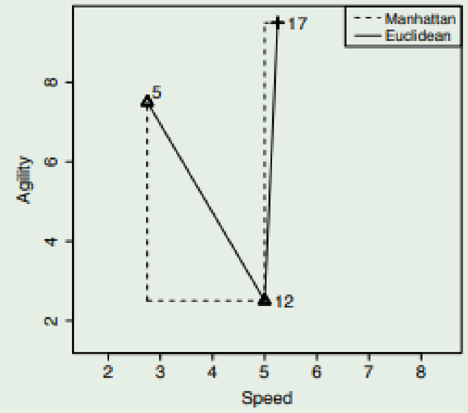
- The Euclidean and Manhattan distances are special cases of Minkowski distance
- The Minkowski distance between two instances **a** and **b** in a feature space with m descriptive features is:

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

  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).

Nearest Neighbor

The Nearest Neighbour Algorithm

Require: set of training instances
Require: a query to be classified

1: Iterate across the instances in memory and find the instance that is shortest distance from the query position in the feature space.
2: Make a prediction for the query equal to the value of the target feature of the nearest neighbor.

Nearest Neighbor

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

Example

- Should we draft an athlete with the following profile:

SPEED= 6.75, AGILITY= 3

Nearest Neighbor

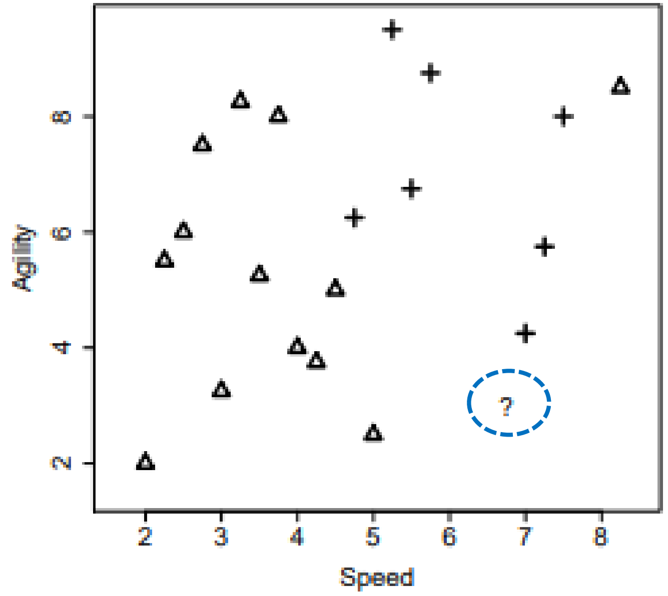


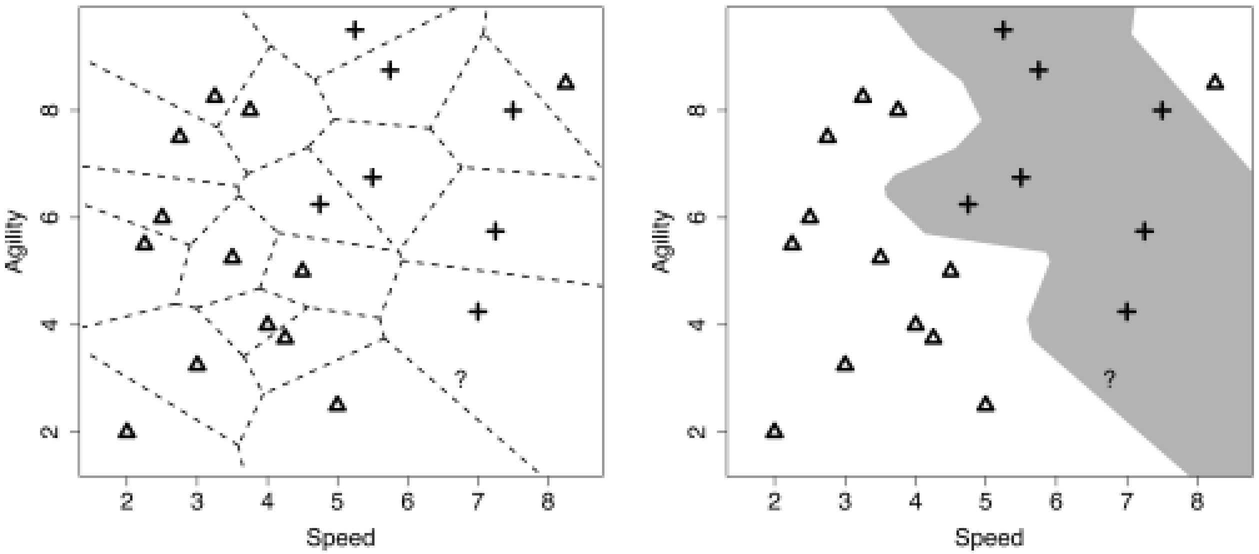
Figure: A feature space plot of the data in Table 2 [25] with the position in the feature space of the query represented by the ? marker. The triangles represent *'Non-draft'* instances and the crosses represent the *'Draft'* instances.

Nearest Neighbor

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

Nearest Neighbor



(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.

Handling Noisy Data

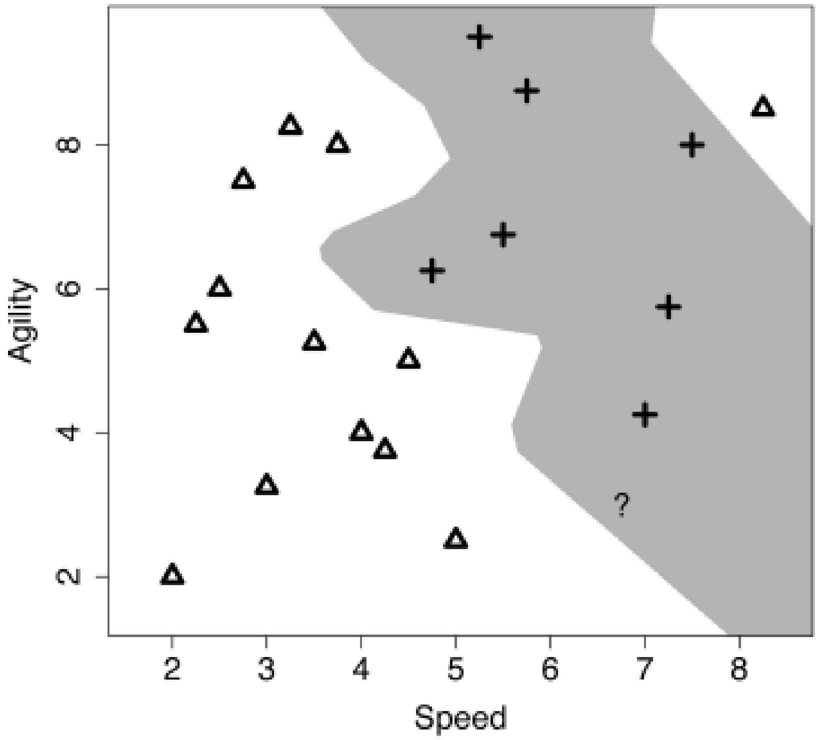


Figure: Is the instance at the top right of the diagram really *noise*?

Nearest Neighbor

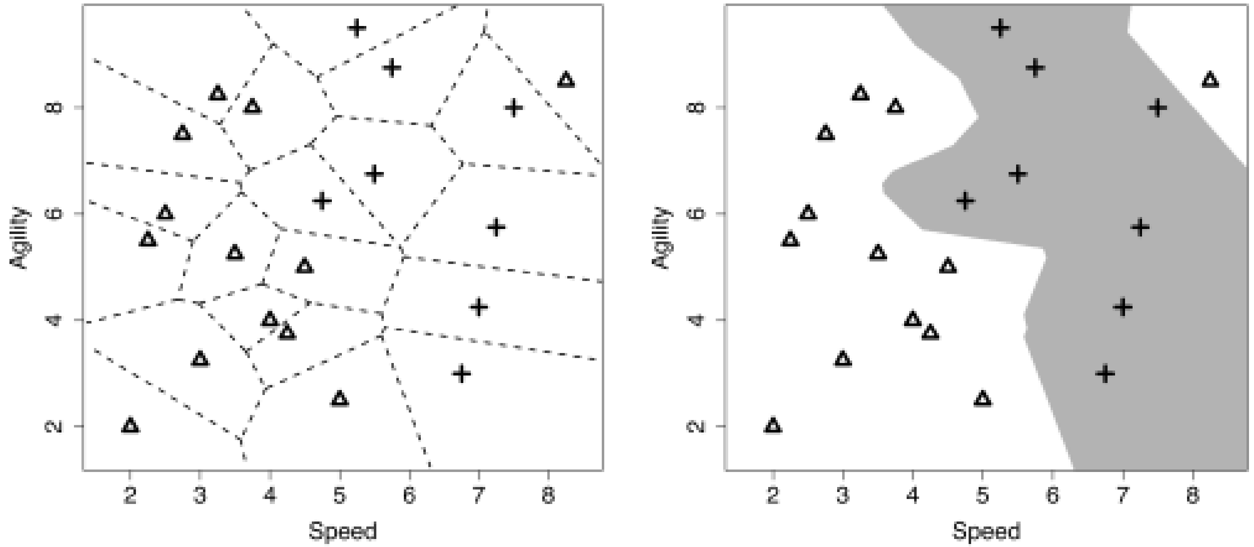
- One of the great things about nearest neighbor algorithm is that we can add new data to update the model very easily.

Nearest Neighbor

Table: The extended version of the college athletes dataset.

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

Nearest Neighbor



(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.

Handling Noisy Data

K-Nearest Neighbors

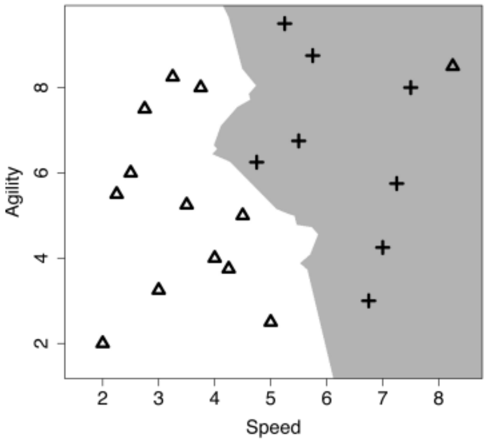
- The k nearest neighbors model predicts the target level with the majority vote from the set of k nearest neighbors to the query **q**:

$$\mathbb{M}_k(\mathbf{q}) = \underset{l \in levels(t)}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(t_i, l)$$

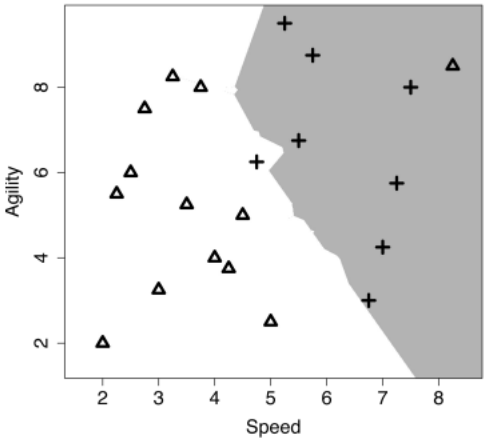
THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Handling Noisy Data

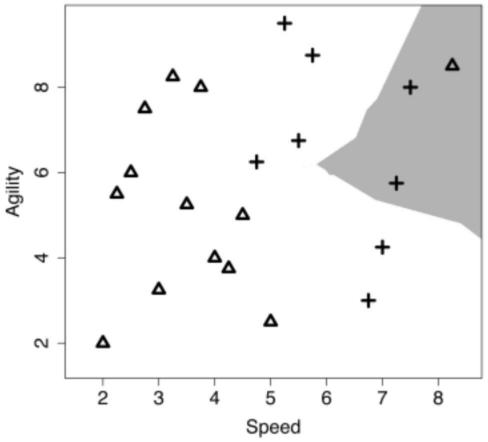
K-Nearest Neighbors



(a) k = 3



(a) k = 5



(a) k = 15

Figure: The decision boundary using majority classification of the k-nearest neighbors.

Data Normalization

Table: A dataset listing the salary and age information for customers and whether or not the purchased a pension plan .

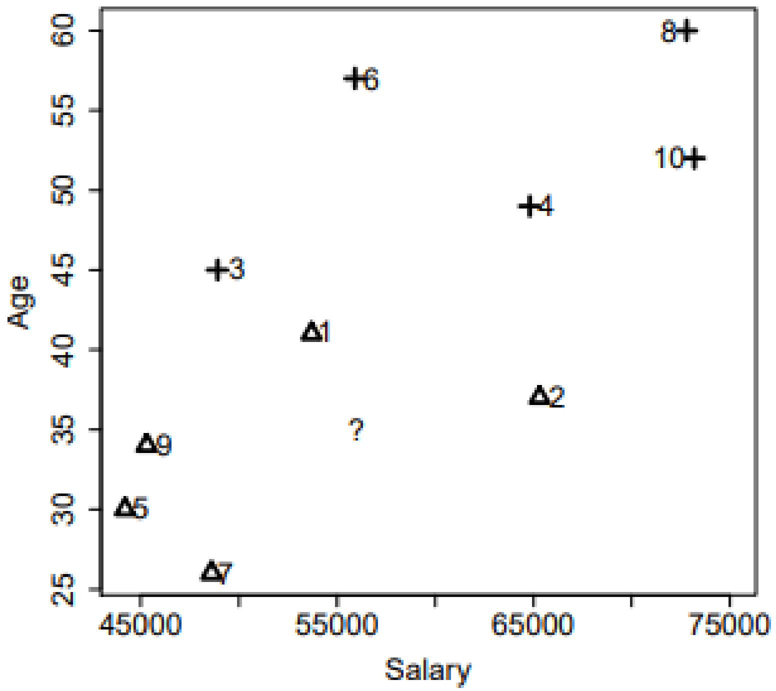
ID	Salary	Age	Purchased
1	53700	41	No
2	65300	37	No
3	48900	45	Yes
4	64800	49	Yes
5	44200	30	No
6	55900	57	Yes
7	48600	26	No
8	72800	60	Yes
9	45300	34	No
10	73200	52	Yes

- The marketing department wants to decide whether or not they should contact a customer with the following profile:

$$\langle \text{SALARY} = 56,000, \text{AGE} = 35 \rangle$$

Data Normalization

- Figure: The salary and age feature space with the data in the customer dataset plotted.
- The instances are labelled their IDs, triangles represent the negative instances and crosses represent the positive instances.
- The location of the query [SALARY = 56000, AGE = 35] is indicated by the ?.



Data Normalization

				Salary and Age		Salary Only		Age Only	
ID	Salary	Age	Purch.	Dist.	Neigh.	Dist.	Neigh.	Dist.	Neigh.
1	53700	41	No	2300.0078	2	2300	2	6	4
2	65300	37	No	9300.0002	6	9300	6	2	2
3	48900	45	Yes	7100.0070	3	7100	3	10	6
4	64800	49	Yes	8800.0111	5	8800	5	14	7
5	44200	30	No	11800.0011	8	11800	8	5	5
6	55900	57	Yes	102.3914	1	100	1	22	9
7	48600	26	No	7400.0055	4	7400	4	9	3
8	72800	60	Yes	16800.0186	9	16800	9	25	10
9	45300	34	No	10700.0000	7	10700	7	1	1
10	73200	52	Yes	17200.0084	10	17200	10	17	8

Data Normalization

- This odd prediction is caused by features taking different ranges of values, this is equivalent to features having different variances.
- We can adjust for this using normalization using the range normalization equation as follows:

$$a_i' = \frac{a_i - min(a)}{max(a) - min(a)} \times (high - low) + low$$

Data Normalization

	Normalized Dataset			Salary and Age		Salary Only		Age Only	
ID	Salary	Age	Purch.	Dist.	Neigh.	Dist.	Neigh.	Dist.	Neigh.
1	0.3276	0.4412	No	0.1935	1	0.0793	2	0.17647	4
2	0.7276	0.3235	No	0.3260	2	0.3207	6	0.05882	2
3	0.1621	0.5588	Yes	0.3827	5	0.2448	3	0.29412	6
4	0.7103	0.6765	Yes	0.5115	7	0.3034	5	0.41176	7
5	0.0000	0.1176	No	0.4327	6	0.4069	8	0.14706	3
6	0.4034	0.9118	Yes	0.6471	8	0.0034	1	0.64706	9
7	0.1517	0.0000	No	0.3677	3	0.2552	4	0.26471	5
8	0.9862	1.0000	Yes	0.9361	10	0.5793	9	0.73529	10
9	0.0379	0.2353	No	0.3701	4	0.3690	7	0.02941	1
10	1.0000	0.7647	Yes	0.7757	9	0.5931	10	0.50000	8

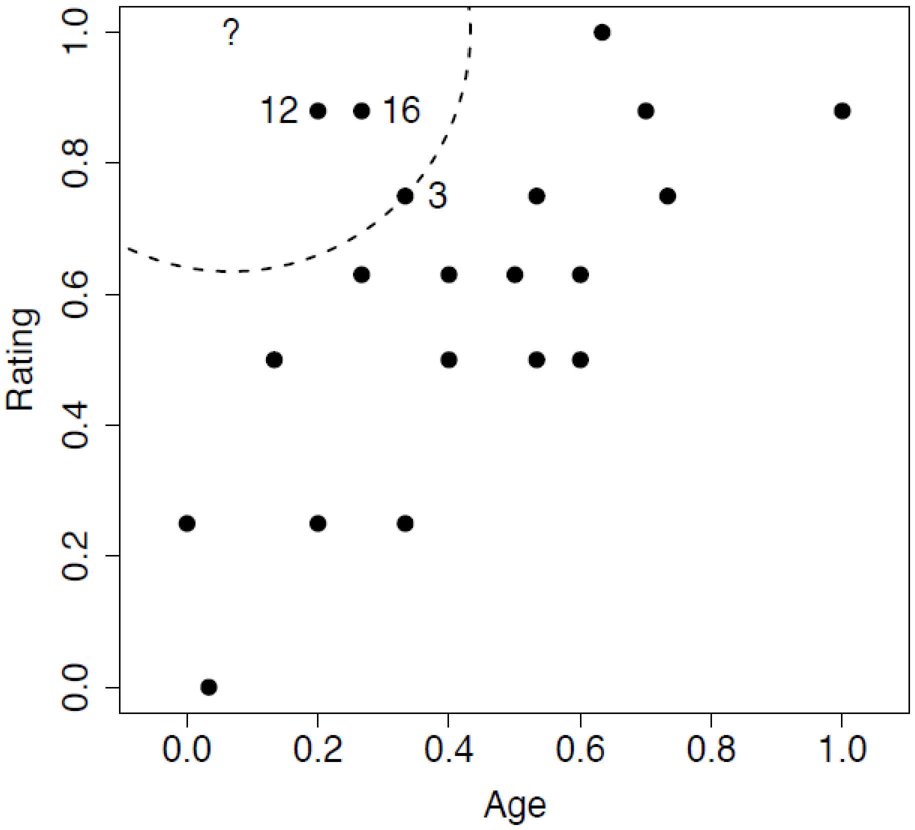
Data Normalization

Normalizing the data is an important thing to do for almost all machine learning algorithms, not just the nearest neighbor!

Predicting Continuous Targets

- Return the average value in the neighborhood:

$$\mathbb{M}_k(\mathbf{q}) = \frac{1}{k}\sum_{i=1}^{k} t_i$$



Other Measures of Similarity

- **Russel-Rao similarity** is defined as the ratio between the number of co-presences and the total number of binary features.
- **Sokal-Michener similarity** is defined as the ratio between the total number of co-presences and co-absences, and the total number of binary features considered.
- **Jacquard index** ignores co-absences
- **Cosine similarity** between two instances is the cosine of the inner angle between the two vectors that extend from the origin to each instance.
- **Mahalanobis distance** uses covariance to scale distances so that distances along a direction where the dataset is spread out a lot are scaled down and distances along directions where the dataset is tightly packed are scaled up.

Co-Occurrences

Table: A binary dataset listing the behavior of two individuals on a website during a trial period and whether or not they subsequently signed-up for the website.

ID	Profile	FAQ	Help Forum	Newsletter	Liked	Signup
1	1	1	1	0	1	Yes
2	1	0	0	0	0	No

Co-Occurrences

Who is q more similar to d_1 or d_2?

$$\mathbf{q} = \langle \textsc{Profile}{:}1, \textsc{FAQ}{:}0, \textsc{Help Forum}{:}1, \textsc{Newsletter}{:}0, \textsc{Liked}{:}0, \rangle$$

ID	Profile	FAQ	Help Forum	Newsletter	Liked	Signup
1	1	1	1	0	1	Yes
2	1	0	0	0	0	No

Co-Occurrences

		q Pres.	q Abs.
$\mathbf{d}_1$	Pres.	CP=2	PA=0
	Abs.	AP=2	CA=1

		q Pres.	q Abs.
$\mathbf{d}_2$	Pres.	CP=1	PA=1
	Abs.	AP=0	CA=3

Table: The similarity between the current trial user, **q**, and the two users in the dataset, $\mathbf{d}_1$ and $\mathbf{d}_2$, in terms of co-presence (CP), co-absence (CA), presence-absence (PA), and absence-presence (AP).

Co-Occurrences

Russel-Rao

$$sim_{RR}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|}$$

Example

$$sim_{RR}(\mathbf{q}, \mathbf{d}_1) = \frac{2}{5} = 0.4$$
$$sim_{RR}(\mathbf{q}, \mathbf{d}_2) = \frac{1}{5} = 0.2$$

- The current trial user is judged to be more similar to instance $\mathbf{d}_1$ then $\mathbf{d}_2$.

Co-Occurrences

Sokal-Michener

$$sim_{SM}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d}) + CA(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|}$$

Example

$$sim_{SM}(\mathbf{q}, \mathbf{d}_1) = \frac{3}{5} = 0.6$$

$$sim_{SM}(\mathbf{q}, \mathbf{d}_2) = \frac{4}{5} = 0.8$$

- The current trial user is judged to be more similar to instance $\mathbf{d}_2$ then $\mathbf{d}_1$.

Co-Occurrences

Jaccard

$$sim_J(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{CP(\mathbf{q}, \mathbf{d}) + PA(\mathbf{q}, \mathbf{d}) + AP(\mathbf{q}, \mathbf{d})}$$

Example

$$sim_J(\mathbf{q}, \mathbf{d}_1) = \frac{2}{4} = 0.5$$

$$sim_J(\mathbf{q}, \mathbf{d}_2) = \frac{1}{2} = 0.5$$

- The current trial user is judged to be equally similar to instance $\mathbf{d}_1$ and $\mathbf{d}_2$!

Cosine Similarity

- Cosine similarity between two instances is the cosine of the inner angle between the two vectors that extend from the origin to each instance.

Cosine

$$sim_{COSINE}(\mathbf{a}, \mathbf{b}) = \frac{(\mathbf{a}[1] \times \mathbf{b}[1]) + \cdots + (\mathbf{a}[m] \times \mathbf{b}[m])}{\sqrt{\sum_{i=1}^{m} \mathbf{a}[i]^2} \times \sqrt{\sum_{i=1}^{m} \mathbf{b}[i]^2}}$$

Cosine Similarity

- Calculate the cosine similarity between the following two instances:

$$\mathbf{d}_1 = \langle \text{SMS} = 97, \text{VOICE} = 21 \rangle$$

$$\mathbf{d}_2 = \langle \text{SMS} = \cancel{18}, \text{VOICE} = 184 \rangle \qquad \text{SMS} = 181$$

$$sim_{COSINE}(\mathbf{d}_1, \mathbf{d}_1) = \frac{(97 \times 181) + (21 \times 184)}{\sqrt{97^2 + 21^2} \times \sqrt{181^2 + 184^2}}$$

$$= 0.8362$$

Cosine Similarity

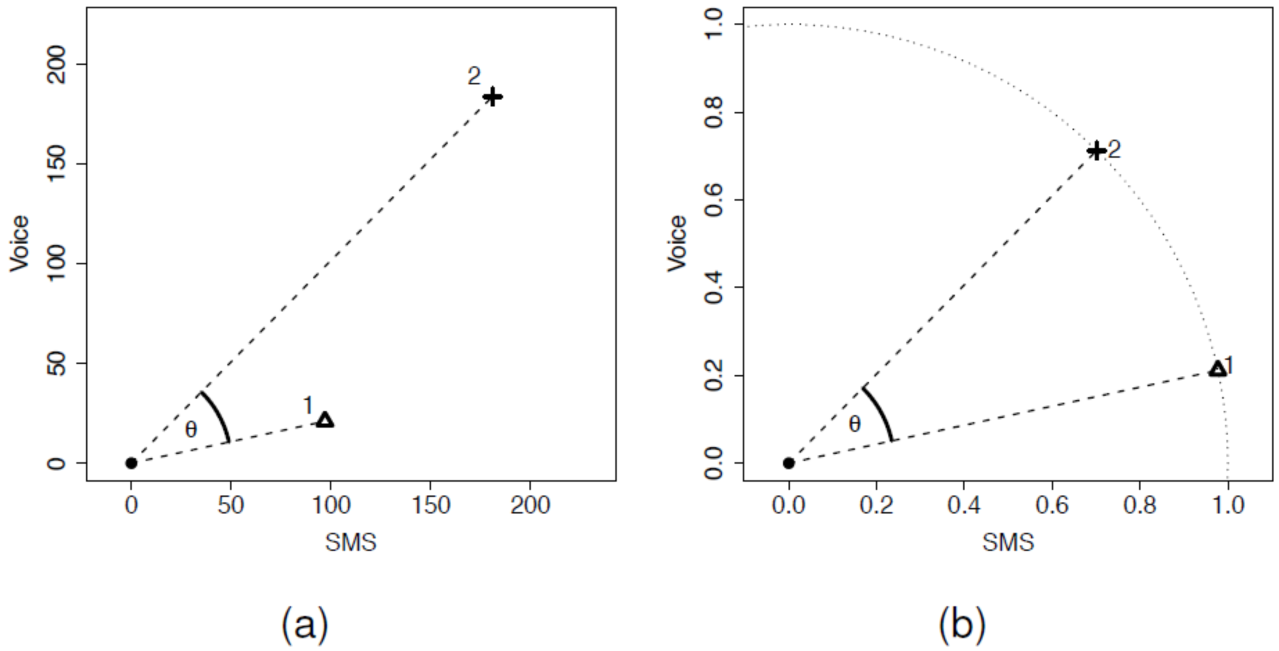


Figure: (a) The θ represents the inner angle between the vector emanating from the origin to instance $\mathbf{d}_1$ $\langle \text{SMS} = 97, \text{VOICE} = 21 \rangle$ and the vector emanating from the origin to instance $\mathbf{d}_2$ $\langle \text{SMS} = 181, \text{VOICE} = 184 \rangle$; (b) shows $\mathbf{d}_1$ and $\mathbf{d}_2$ normalized to the unit circle.

Cosine Similarity

- Calculate the cosine similarity between the following two instances:

$$\mathbf{d}_1 = \langle \text{SMS} = 97, \text{VOICE} = 21 \rangle$$

$$\mathbf{d}_3 = \langle \text{SMS} = 194, \text{VOICE} = 42 \rangle$$

$$sim_{COSINE}(\mathbf{d}_1, \mathbf{d}_1) = \frac{(97 \times 194) + (21 \times 42)}{\sqrt{97^2 + 21^2} \times \sqrt{194^2 + 42^2}}$$

$$= 1$$

Mahalanobis Distance

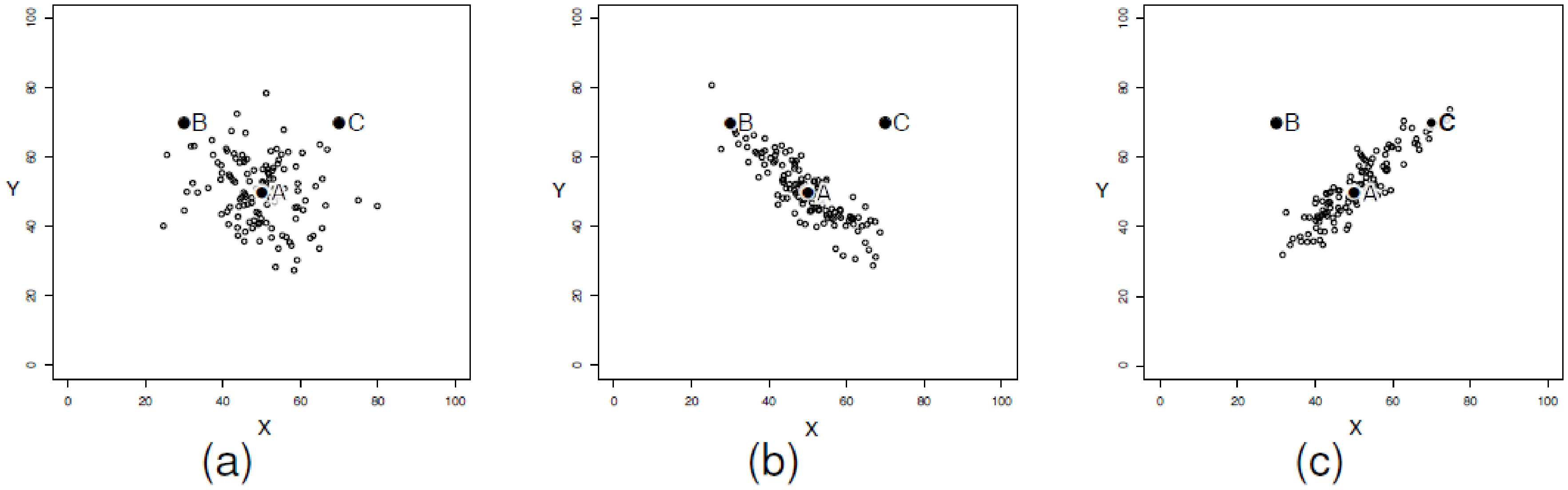


Figure: Scatter plots of three bivariate datasets with the same center point A and two queries B and C both equidistant from A. (a) A dataset uniformly spread around the center point. (b) A dataset with negative covariance. (c) A dataset with positive covariance.

Mahalanobis Distance

- The mahalanobis distance uses covariance to scale distances so that distances along a direction where the dataset is spreadout a lot are scaled down and distances along directions where the dataset is tightly packed are scaled up.

$$Mahalanobis(\mathbf{a}, \mathbf{b}) =$$

$$[\mathbf{a}[1] - \mathbf{b}[1], \ldots, \mathbf{a}[m] - \mathbf{b}[m]] \times \sum^{-1} \times \begin{bmatrix} \mathbf{a}[1] - \mathbf{b}[1] \\ \ldots \\ \mathbf{a}[m] - \mathbf{b}[m] \end{bmatrix}$$

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.

Summary

- Nearest neighbor models are very sensitive to noise in the target feature the easiest way to solve this problem is to employ a k nearest neighbor.
- Normalization techniques should almost always be applied when nearest neighbor models are used. It is easy to adapt a nearest neighbor model to continuous targets.
- There are many different measures of similarity.
- As the number of instances becomes large, a nearest neighbor model will become slower—techniques such as the k-d tree can help with this issue.
- Feature selection is a particularly important process for nearest neighbor algorithms it alleviates the curse of dimensionality.