Similarity-based Learning
Sections 5.4, 5.5

John D. Kelleher and Brian Mac Namee and Aoife D'Arcy

Handling Noisy Data

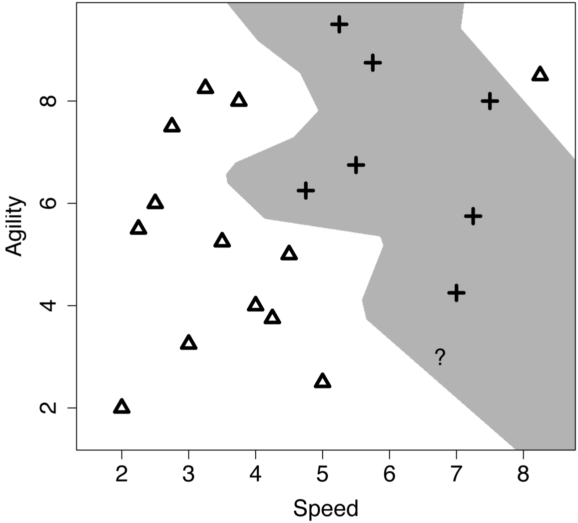


Figure: Is the instance at the top right of the diagram really *noise*?

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

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

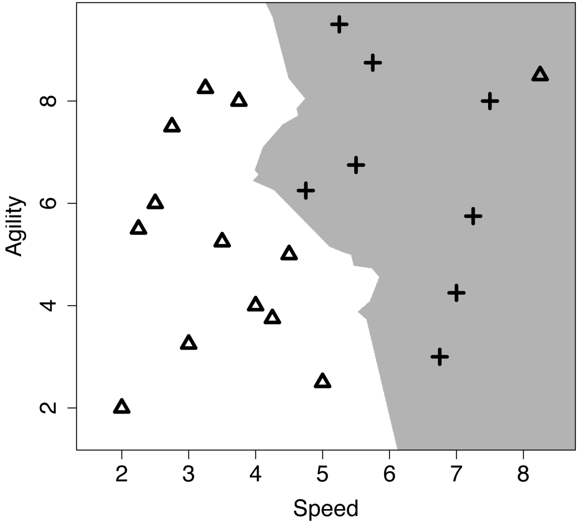


Figure: The decision boundary using majority classification of the nearest 3 neighbors.

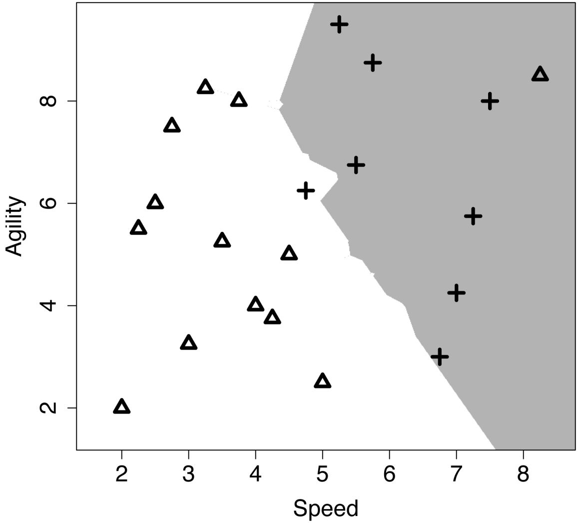


Figure: The decision boundary using majority classification of the nearest 5 neighbors.

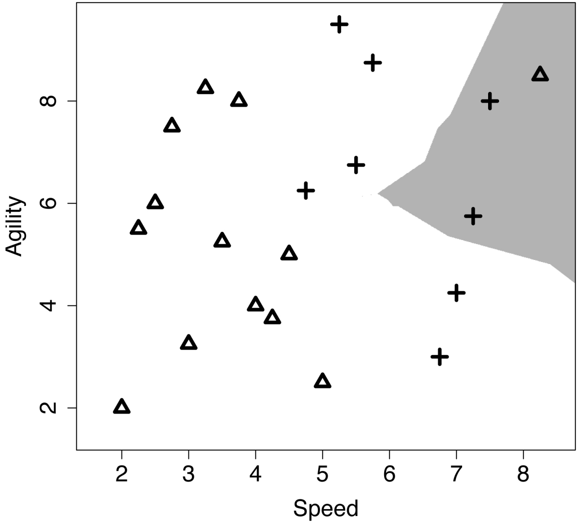


Figure: The decision boundary when *k* is set to 15.

- In a distance weighted k nearest neighbor algorithm the contribution of each neighbor to the classification decision is weighted by the reciprocal of the squared distance between the neighbor **d** and the query **q**:

$$\frac{1}{dist(\mathbf{q},\mathbf{d})^2} \quad (2)$$

- The weighted k nearest neighbor model is defined as:

$$\mathbb{M}_k(\mathbf{q}) = \underset{l \in levels(t)}{\operatorname{argmax}} \sum_{i=1}^{k} \frac{1}{dist(\mathbf{q},\mathbf{d_i})^2} \times \delta(t_i, l) \quad (3)$$

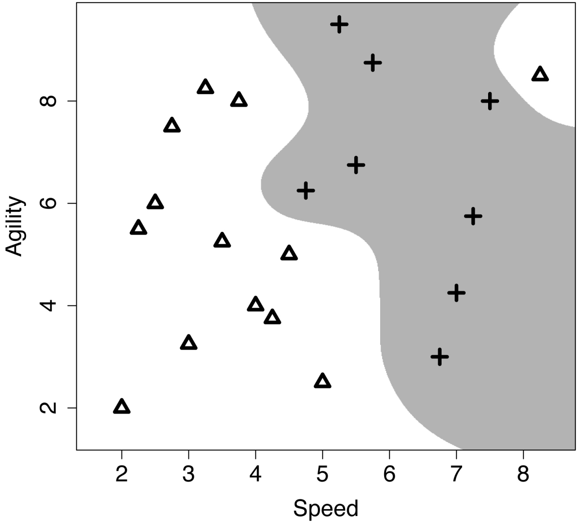


Figure: The weighted *k* nearest neighbor model decision boundary.

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:

$$\langle \textsc{Salary} = 56,000, \textsc{Age} = 35 \rangle$$

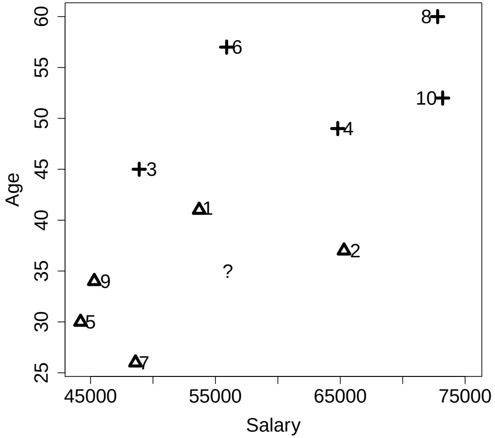


Figure: The salary and age feature space with the data in Table 1 [12] plotted. The instances are labelled their IDs, triangles represent the negative instances and crosses represent the positive instances. The location of the query $\langle \text{SALARY} = 56,000, \text{AGE} = 35 \rangle$ is indicated by the ?.

				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

- 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; the equation for range normalization is:

$$a_i^{'} = \frac{a_i - min(a)}{max(a) - min(a)} \times (high - low) + low \tag{4}$$

	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

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

Predicting Continuous Targets

- Return the average value in the neighborhood:

$$\mathbb{M}_k(\mathbf{q}) = \frac{1}{k}\sum_{i=1}^{k} t_i \tag{5}$$

Table: A dataset of whiskeys listing the age (in years) and the rating (between 1 and 5, with 5 being the best) and the bottle price of each whiskey.

ID	Age	Rating	Price	ID	Age	Rating	Price
1	0	2	30.00	11	19	5	500.00
2	12	3.5	40.00	12	6	4.5	200.00
3	10	4	55.00	13	8	3.5	65.00
4	21	4.5	550.00	14	22	4	120.00
5	12	3	35.00	15	6	2	12.00
6	15	3.5	45.00	16	8	4.5	250.00
7	16	4	70.00	17	10	2	18.00
8	18	3	85.00	18	30	4.5	450.00
9	18	3.5	78.00	19	1	1	10.00
10	16	3	75.00	20	4	3	30.00

Table: The whiskey dataset after the descriptive features have been normalized.

ID	Age	Rating	Price	ID	Age	Rating	Price
1	0.0000	0.25	30.00	11	0.6333	1.00	500.00
2	0.4000	0.63	40.00	12	0.2000	0.88	200.00
3	0.3333	0.75	55.00	13	0.2667	0.63	65.00
4	0.7000	0.88	550.00	14	0.7333	0.75	120.00
5	0.4000	0.50	35.00	15	0.2000	0.25	12.00
6	0.5000	0.63	45.00	16	0.2667	0.88	250.00
7	0.5333	0.75	70.00	17	0.3333	0.25	18.00
8	0.6000	0.50	85.00	18	1.0000	0.88	450.00
9	0.6000	0.63	78.00	19	0.0333	0.00	10.00
10	0.5333	0.50	75.00	20	0.1333	0.50	30.00

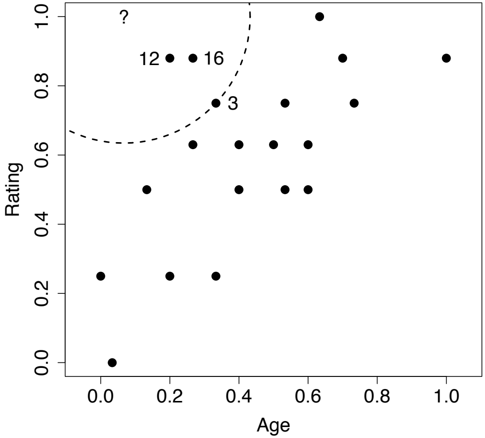


Figure: The AGE and RATING feature space for the whiskey dataset. The location of the query instance is indicated by the ? symbol. The circle plotted with a dashed line demarcates the border of the neighborhood around the query when $k = 3$. The three nearest neighbors to the query are labelled with their ID values.

- The model will return a price prediction that is the average price of the three neighbors:

$$\frac{200.00 + 250.00 + 55.00}{3} = 168.33$$

- In a weighted k nearest neighbor the model prediction equation is changed to:

$$\mathbb{M}_k(\mathbf{q}) = \frac{\displaystyle\sum_{i=1}^{k} \frac{1}{dist(\mathbf{q}, \mathbf{d}_i)^2} \times t_i}{\displaystyle\sum_{i=1}^{k} \frac{1}{dist(\mathbf{q}, \mathbf{d}_i)^2}} \tag{6}$$

Table: The calculations for the weighted k nearest neighbor prediction

ID	Price	Distance	Weight	Price $\times$ Weight
1	30.00	0.7530	1.7638	52.92
2	40.00	0.5017	3.9724	158.90
3	55.00	0.3655	7.4844	411.64
4	550.00	0.6456	2.3996	1319.78
5	35.00	0.6009	2.7692	96.92
6	45.00	0.5731	3.0450	137.03
7	70.00	0.5294	3.5679	249.75
8	85.00	0.7311	1.8711	159.04
9	78.00	0.6520	2.3526	183.50
10	75.00	0.6839	2.1378	160.33
11	500.00	0.5667	3.1142	1557.09
12	200.00	0.1828	29.9376	5987.53
13	65.00	0.4250	5.5363	359.86
14	120.00	0.7120	1.9726	236.71
15	12.00	0.7618	1.7233	20.68
16	250.00	0.2358	17.9775	4494.38
17	18.00	0.7960	1.5783	28.41
18	450.00	0.9417	1.1277	507.48
19	10.00	1.0006	0.9989	9.99
20	30.00	0.5044	3.9301	117.90
		Totals:	99.2604	16 249.85

Other Measures of Similarity

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

Who is q more similar to d_1 or d_2?

$$\mathbf{q} = \langle \text{PROFILE:1}, \text{FAQ:0}, \text{HELP FORUM:1}, \text{NEWSLETTER:0}, \text{LIKED:0}, \rangle$$

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

		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).

Russel-Rao

- The ratio between the number of co-presenses and the total number of binary features considered.

$$sim_{RR}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|} \tag{7}$$

Russel-Rao

$$sim_{RR}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|} \tag{8}$$

$\langle$PROFILE:1, FAQ:0, HELP FORUM:1, NEWSLETTER:0, LIKED:0, $\rangle$

		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

Russel-Rao

$$sim_{RR}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|} \tag{9}$$

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 $\mathbf{d}_2$.

Sokal-Michener

- Sokal-Michener is defined as the ratio between the total number of co-presences and co-absences, and the total number of binary features considered.

$$sim_{SM}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d}) + CA(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|} \quad (10)$$

Sokal-Michener

$$sim_{SM}(\mathbf{q},\mathbf{d}) = \frac{CP(\mathbf{q},\mathbf{d}) + CA(\mathbf{q},\mathbf{d})}{|\mathbf{q}|} \tag{11}$$

$\langle$PROFILE:1, FAQ:0, HELP FORUM:1, NEWSLETTER:0, LIKED:0, $\rangle$

		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

Sokal-Michener

$$sim_{SM}(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d}) + CA(\mathbf{q}, \mathbf{d})}{|\mathbf{q}|} \tag{12}$$

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 $\mathbf{d}_1$.

Jaccard

- The Jacquard index ignore co-absences

$$sim_J(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{CP(\mathbf{q}, \mathbf{d}) + PA(\mathbf{q}, \mathbf{d}) + AP(\mathbf{q}, \mathbf{d})} \tag{13}$$

Jaccard

$$sim_J(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{CP(\mathbf{q}, \mathbf{d}) + PA(\mathbf{q}, \mathbf{d}) + AP(\mathbf{q}, \mathbf{d})} \tag{14}$$

$\langle$ PROFILE:1, FAQ:0, HELP FORUM:1, NEWSLETTER:0, LIKED:0, $\rangle$

		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

Jaccard

$$sim_J(\mathbf{q}, \mathbf{d}) = \frac{CP(\mathbf{q}, \mathbf{d})}{CP(\mathbf{q}, \mathbf{d}) + PA(\mathbf{q}, \mathbf{d}) + AP(\mathbf{q}, \mathbf{d})} \tag{15}$$

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

Cosine

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

- Calculate the cosine similarity between the following two instances:

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

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

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

- Calculate the cosine similarity between the following two instances:

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

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

$$\begin{aligned} sim_{COSINE}(\mathbf{d}_1, \mathbf{d}_1) &= \frac{(97 \times 181) + (21 \times 184)}{\sqrt{97^2 + 21^2} \times \sqrt{181^2 + 184^2}} \\ &= 0.8362 \end{aligned}$$

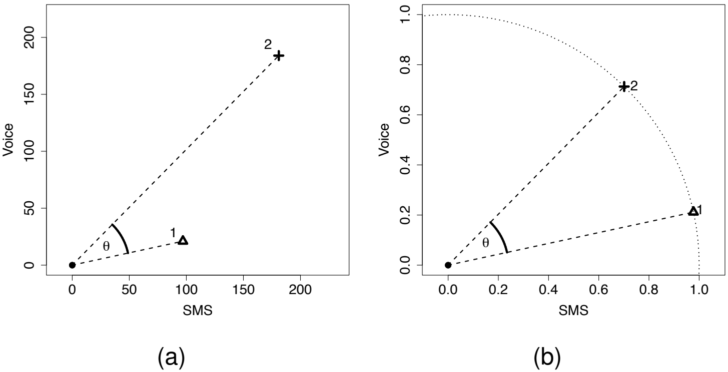


Figure: (a) The θ represents the inner angle between the vector emanating from the origin to instance $\mathbf{d}_1$ $\langle \text{SMS} = 97, \text{VOICE} = 21 \rangle$ and the vector emanating from the origin to instance $\mathbf{d}_2$ $\langle \text{SMS} = 181, \text{VOICE} = 184 \rangle$; (b) shows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

- Calculate the cosine similarity between the following two instances:

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

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

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

- Calculate the cosine similarity between the following two instances:

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

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

$$sim_{COSINE}(\mathbf{d}_1, \mathbf{d}_1) = \frac{(97 \times 194) + (21 \times 42)}{\sqrt{97^2 + 21^2} \times \sqrt{194^2 + 42^2}}$$
$$= 1$$

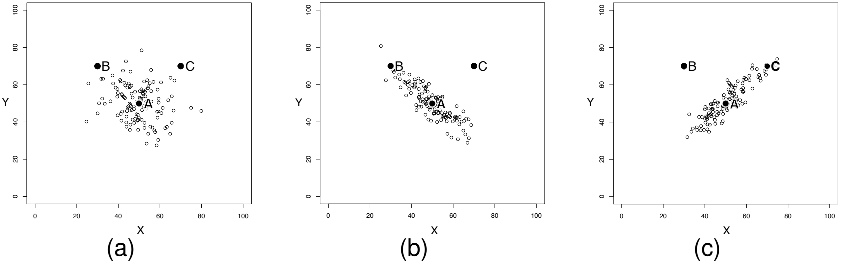


Figure: Scatter plots of three bivariate datasets with the same center point A and two queries B and C both equidistant from A. (a) A dataset uniformly spread around the center point. (b) A dataset with negative covariance. (c) A dataset with positive covariance.

- 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.

$$Mahalanobis(\mathbf{a}, \mathbf{b}) = [\mathbf{a}[1] - \mathbf{b}[1], \ldots, \mathbf{a}[m] - \mathbf{b}[m]] \times \sum^{-1} \times \begin{bmatrix} \mathbf{a}[1] - \mathbf{b}[1] \\ \ldots \\ \mathbf{a}[m] - \mathbf{b}[m] \end{bmatrix} \tag{16}$$

- Similar to Euclidean distance, the mahalanobis distance squares the differences of the features.
- But it also rescales the differences (using the inverse covariance matrix) so that all the features have unit variance and the effects of covariance is removed.

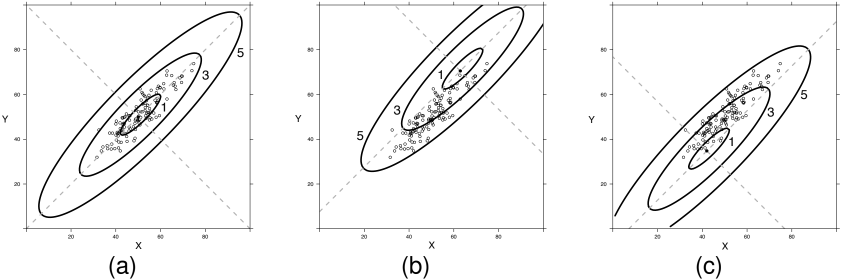


Figure: The coordinate systems defined by the mahalanobis metric using the co-variance matrix for the dataset in Figure 9(c) [46] using three different origins: (a) $(50, 50)$, (b) $(63, 71)$, (c) $(42, 35)$. The ellipses in each figure plot the 1, 3, and 5 unit distances contours from each origin.

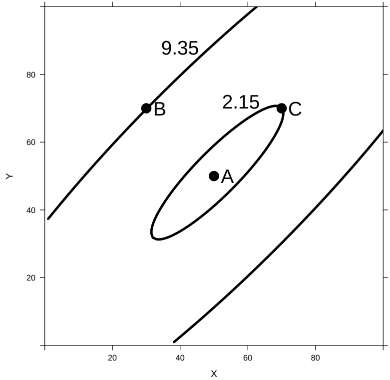


Figure: The effect of using a mahalanobis versus euclidean distance. Point A is the center of mass of the dataset in Figure 9(c) [46]. The ellipses plot the mahalanobis distance contours from A that B and C lie on. In euclidean terms B and C are equidistant from A, however using the mahalanobis metric C is much closer to A than B.

Feature Selection

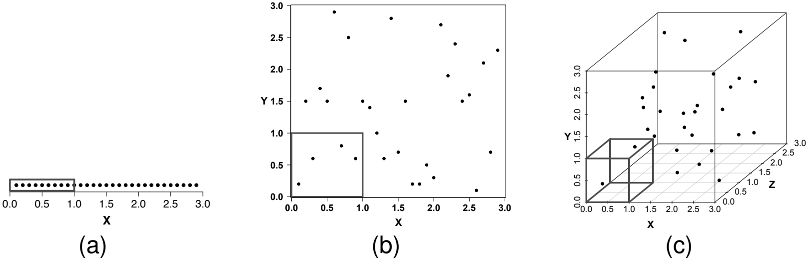


Figure: A set of scatter plots illustrating the curse of dimensionality. Across figures (a), (b) and (c) the density of the marked unit hypercubes decreases as the number of dimensions increases.

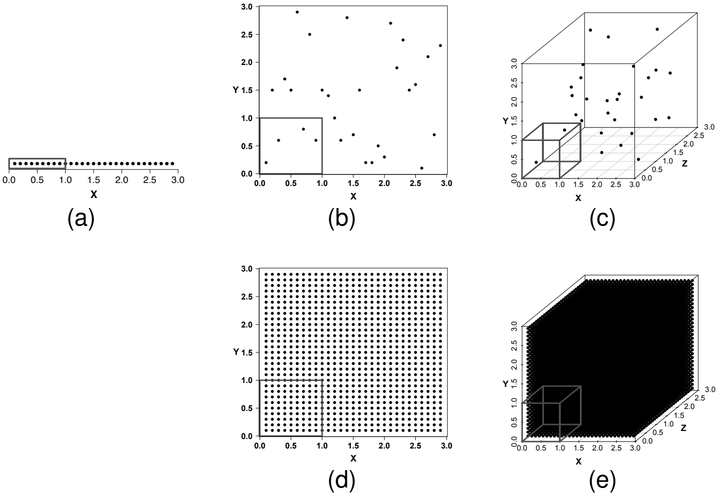


Figure: Figures (d) and (e) illustrate the cost we must incur if we wish to maintain the density of the instances in the feature space as the dimensionality of the feature space increases.

- During our discussion of feature selection approaches it will be useful to distinguish between different classes of descriptive features:
    1. Predictive
    2. Interacting
    3. Redundant
    4. Irrelevant

- When framed as a local search problem feature selection is defined in terms of an iterative process consisting of the following stages:
  1. Subset Generation
  2. Subset Selection
  3. Termination Condition
- The search can move through the search space in a number of ways:
  - Forward sequential selection
  - Backward sequential selection

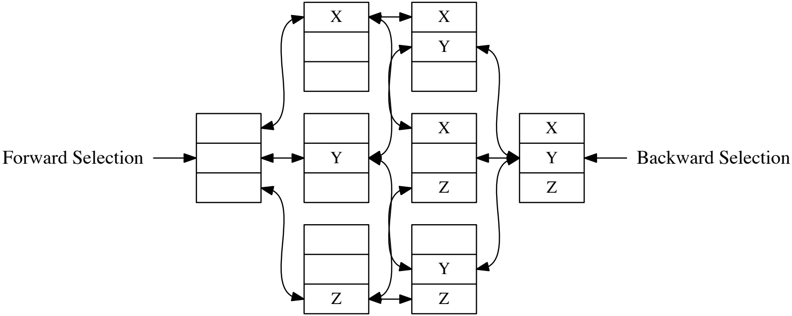
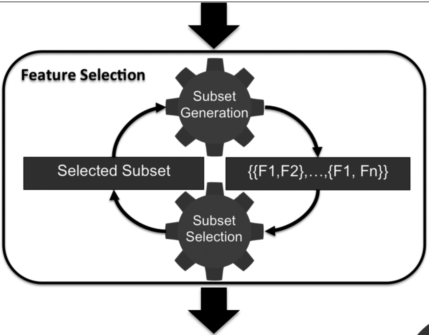


Figure: Feature Subset Space for a dataset with 3 features X, Y, Z.

ID	F1	F2	F3	...	F150	F151	...	F227	F228	...	F387	...	Fn	Target
1	-	-	-	...	-	-	...	-	-	...	-	...	-	-
2	-	-	-	...	-	-	...	-	-	...	-	...	-	-
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
K	-	-	-	...	-	-	...	-	-	...	-	...	-	-



ID	F2	F150	F227	F387	Target
1	-	-	-	-	-
2	-	-	-	-	-
...	...	...	...	...	...
K	-	-	-	-	-

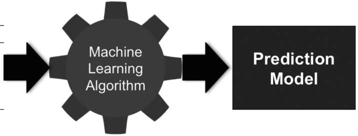


Figure: The process of model induction with feature selection.

Efficient Memory Search

- Assuming that the training set will remain relatively stable it is possible to speed up the prediction speed of a nearest neighbor model by investing in some one-off computation to create an index of the instances that enables efficient retrieval of the nearest neighbors.
- The k-d tree, which is short for k-dimensional tree, is one of the best known of these indexes.

Example

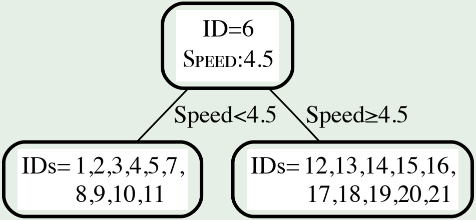
Let's build a k-d tree for the college athlete dataset

Table: The speed and agility ratings for 22 college athletes labelled with the decisions for whether they were drafted or not.

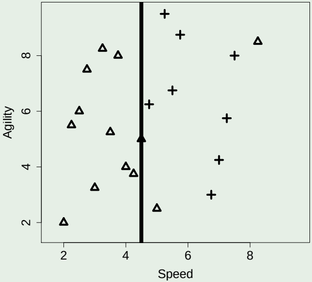
ID	Speed	Agility	Draft	ID	Speed	Agility	Draft
1	2.50	6.00	No	12	5.00	2.50	No
2	3.75	8.00	No	13	8.25	8.50	No
3	2.25	5.50	No	14	5.75	8.75	Yes
4	3.25	8.25	No	15	4.75	6.25	Yes
5	2.75	7.50	No	16	5.50	6.75	Yes
6	4.50	5.00	No	17	5.25	9.50	Yes
7	3.50	5.25	No	18	7.00	4.25	Yes
8	3.00	3.25	No	19	7.50	8.00	Yes
9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

Example

First split on the SPEED feature



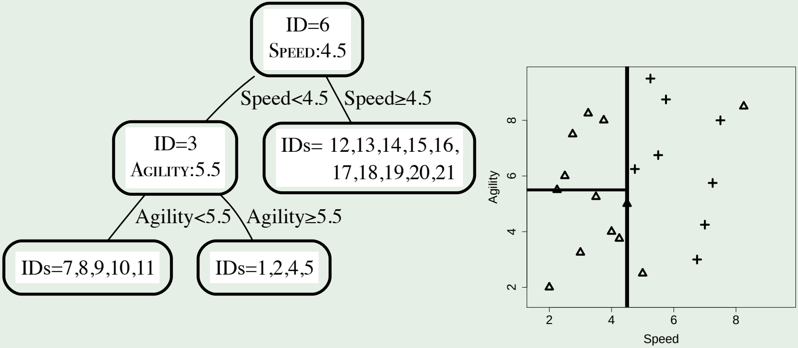
(a)



(b)

Example

Next split on the AGILITY feature



(a) (b)

Figure: (a) the k-d tree after the dataset at the left child of the root has been split using the AGILITY feature with a threshold of 5.5. (b)

Example

After completing the tree building process

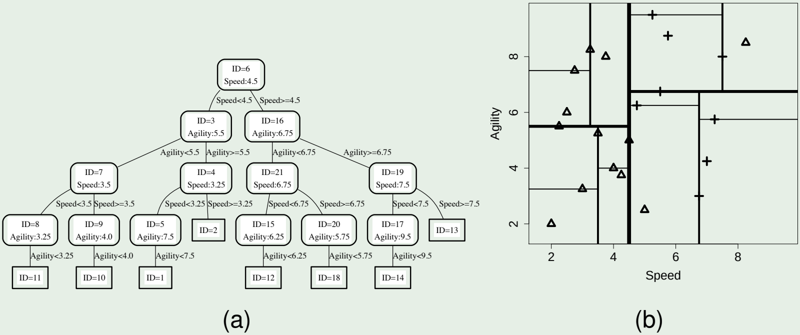
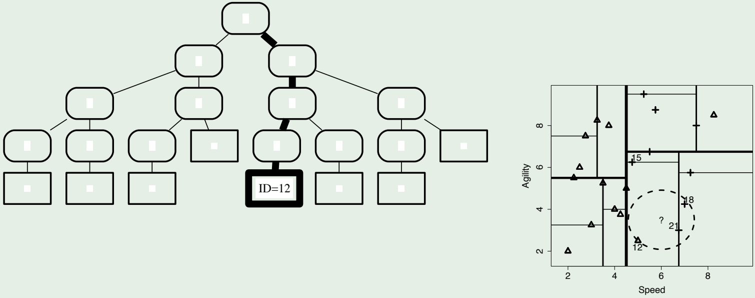


Figure: (a) The complete k-d tree generated, for the dataset in Table 7 [60]. (b) the partitioning of the feature space by the k-d tree in (a)

Example

Finding neighbors for query SPEED= 6, AGILITY= 3.5.



(a) (b)

Figure: (a) the path taken from the root node to a leaf node when we

Example

Finding neighbors for query SPEED= 6, AGILITY= 3.5.

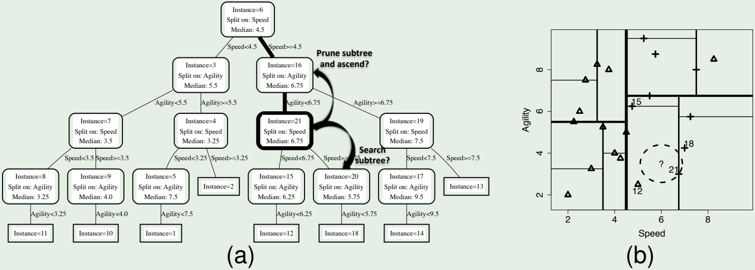
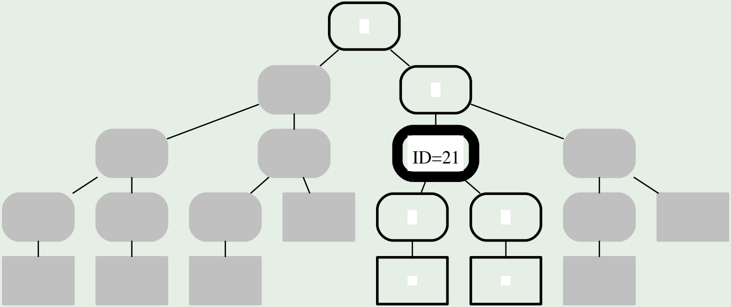


Figure: (a) the state of the retrieval process after instance $\mathbf{d}_{21}$ has been stored as *current-best*. (b) the dashed circle illustrates the extent of the target hypersphere after *current-best-distance* has been updated.

Example



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**.
- **Feature selection** is a particularly important process for nearest neighbor algorithms it alleviates the **curse of dimensionality**.
- 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.