CS 5525
Solutions to Homework Assignment 2

Meghendra Singh

September 27, 2016

[10] 1.

Given that the sample covariance between users height and the time being a member of the community is **C** which is a positive number. Here height is specified in mm and the the time being a member of the community is given in years.

(i) If height is re-defined as height above the average (mean), the covariance **C** would remain the same. A linear transformation changes the covariance between two vectors only if the transformation multiplies (or divides) the vectors by some constant other than 1. Addition (or subtraction) of a constant from the vectors does not change the covariance between them. Redefining height as height above average is equivalent to subtracting a constant (the mean value) from all the elements of the height attribute. Therefore, the covariance between the redefined "height above average" and "years being a member" vector should remain equal to the earlier covariance **C**. The following proves this conclusion:

Let, H and Y represent the vectors for the original height of members and years being members, respectively. Therefore, the covariance between H and Y is given by:

$$cov(H,Y) = \frac{\sum_{i=1}^{n}(H_i - \overline{H})(Y_i - \overline{Y})}{n-1} = \mathbf{C}$$

Given that, upon redefinition,

$$H_{new} = H - \overline{H} \tag{1}$$

and,

$$\overline{H_{new}} = 0 \tag{2}$$

The new covariance is given by,

$$cov(H_{new},Y) = \frac{\sum_{i=1}^{n}(H_{new\ i} - \overline{H_{new}})(Y_i - \overline{Y})}{n-1} \tag{3}$$

By (1), (2) and (3) we have,

$$\begin{aligned} cov(H_{new},Y) &= \frac{\sum_{i=1}^{n}((H_i - \overline{H}) - 0)(Y_i - \overline{Y})}{n-1} \\ &= \frac{\sum_{i=1}^{n}(H_i - \overline{H})(Y_i - \overline{Y})}{n-1} \\ &= cov(H,Y) \\ &= \mathbf{C} \end{aligned}$$

Therefore, the redefined height attribute, does not change the covariance **C** between the two attributes.

(ii) If the measurement unit for height is converted from mm to inches (1 inch = 25.4 mm), the covariance will reduce because this linear transformation involves dividing the height attribute by a constant value (25.4). Therefore the new covariance would be smaller than the **C** (the original covariance) and equal to $\frac{\mathbf{C}}{\mathbf{25.4}}$. The following proves this conclusion:

$$cov(H,Y) = \frac{\sum_{i=1}^{n}(H_i - \overline{H})(Y_i - \overline{Y})}{n-1} = \mathbf{C}$$

Given that upon change of measurement unit,

$$H_{new} = \frac{H}{25.4} \tag{1}$$

and,

$$\overline{H_{new}} = \frac{\overline{H_{new}}}{25.4} \tag{2}$$

The new covariance is given by,

$$cov(H_{new},Y) = \frac{\sum_{i=1}^{n}(H_{new\ i} - \overline{H_{new}})(Y_i - \overline{Y})}{n-1} \tag{3}$$

By (1), (2) and (3) we have,

$$\begin{aligned}
cov(H_{new},Y) &= \frac{\sum_{i=1}^{n}(\frac{H_i}{25.4} - \frac{\overline{H}}{25.4})(Y_i - \overline{Y})}{n-1} \\
&= \frac{\sum_{i=1}^{n}\frac{1}{25.4}(H_i - \overline{H})(Y_i - \overline{Y})}{n-1} \\
&= \frac{1}{25.4}\frac{\sum_{i=1}^{n}(H_i - \overline{H})(Y_i - \overline{Y})}{n-1} \\
&= \frac{1}{25.4}cov(H,Y) \\
&= \frac{\mathbf{C}}{25.4}
\end{aligned}$$

Therefore, the change in measurement of height attribute from *mm* to *inches* , reduces the covariance **C** between the two attributes. The new covariance is $\frac{\mathbf{C}}{\mathbf{25.4}}$.

---

[20] 2.

The elements of weight matrix W for a Term-Document matrix are computed as:

$$W_{ij} = tf_{ij} * idf_i = tf_{ij} * \ln\left(\frac{N}{df_i}\right))$$

Where, tf_{ij} is the frequency of term i in document j, N is the total number of documents in the vector space model, df_i is the document frequency of term i (i.e. total number of documents in the vector space model which contain the term i) and idf_i is the inverse document frequency.

(a) Assuming that all documents are of the same length, the term frequency TF matrix is as follows:

	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
D1	0	2	3	4	0	4	0	5
D2	0	3	2	0	4	1	3	0
D3	3	1	5	1	1	3	5	0
D4	2	3	2	0	0	5	3	9

The document frequency DF for each term is given by:

	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
DF	2	4	4	2	2	4	3	2

The inverse document frequency IDF is given by:

	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
IDF	0.693	0	0	0.693	0.693	0	0.287	0.693

The weight matrix, with each element corresponding to $TF * IDF$ comes out to be:

	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
D1	0	0	0	2.772	0	0	0	3.465
D2	0	0	0	0	2.772	0	0.863	0
D3	2.079	0	0	0.693	0.693	0	1.438	0
D4	1.386	0	0	0	0	0	0.863	6.238

(b) The unweighted query **Term4 Term8** (Q) can be specified in term frequency form as:

	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
Q	0	0	0	1	0	0	0	1

The vector space retrieval results in the following cosine similarity scores and rank order for the four documents:

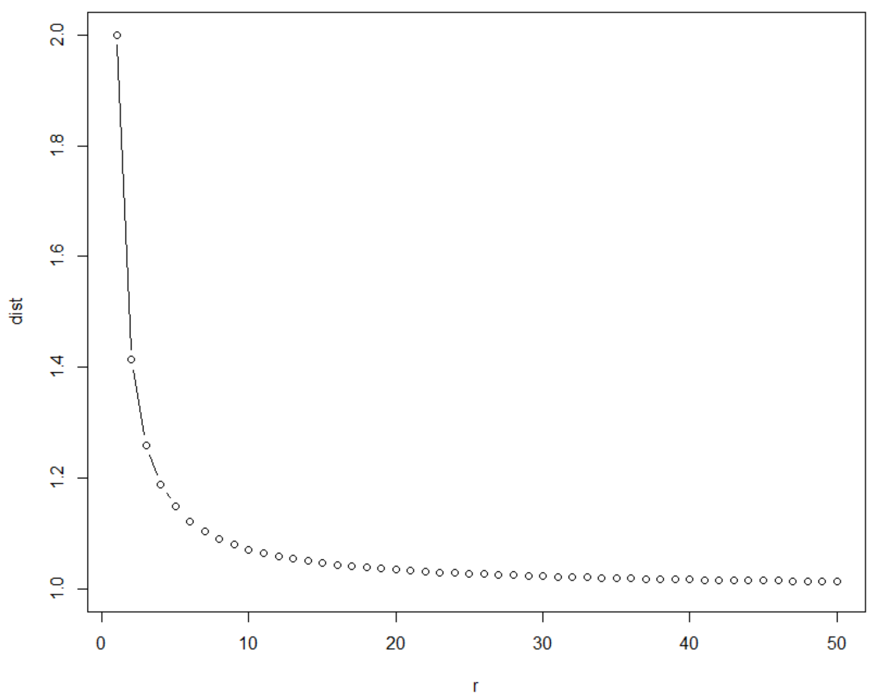
Document	**Cosine Similarity**	**Rank**
$D1$	0.993	1
$D2$	0	4
$D3$	0.180	3
$D4$	0.684	2

[10] 3.

(a) Minkowski distance is a generalized distance metric that can be used to obtain distance between two vectors (points in a normed vector space). For two n dimensional vectors X and Y the Minkowski distance is defined as:

$$dist = \left(\sum_{k=1}^{n} |X_k - Y_k|^r\right)^{\frac{1}{r}} \tag{1}$$

Here, r is the order of the equation and theoretically, the equation can specify an infinite number of distance measures by varying the value of r from 1 to ∞. The following graph shows how the value of Minkowski distance between two points $(0, 0)$ and $(1, 1)$ on the two dimensional coordinate plane changes as r is increased from 1 to 50:



We observe that as r increases from 1 to 50, the distance reduces from 2 to 1.01. At $r = 1$, Minkowski distance is called the Manhattan distance (L_1 distance) which gives the greatest value of distance between two points. As r is increased to 2, the Minkowski distance becomes Euclidean distance or straight line distance (L_2 distance). The L_2 distance value is smaller than L_1 distance value for any two points.

As r approaches ∞, the Minkowski distance becomes the Chebyshev distance (L_∞ or L_{max} distance), which gives the smallest possible distance value between two points. Therefore, as r is increases, the value of distance obtained reduces. $r = 1$ gives the greatest distance value and $r = \infty$ gives the smallest distance value between any two points.

(b) Given x1=0, y1=0 and x2=5, y2=12 are two points on a two-dimensional plane. The values of the Minkowski distance between these two points for different values of r are as follows:

r	**Minkowski Distance**
1	17
2	13
4	12.089
8	12.001

We can observe that as the value of r is increasing, the Minkowski distance reduces. Hence, it can be concluded that for higher values of r, Minkowski distance will keep on reducing.

---

[20] 4.

The Gini Index computations for parent and child nodes when split is made on the Gender and Car Type attributes are as follows:

(i) Gender - On splitting we get two child nodes corresponding to $M(Male)$ and $F(Female)$ categories respectively, with 10 records in each child node. For M category, there are 6 records belonging to class $C0$ and 4 records belonging to class $C1$. For F category, there are 4 records belonging to class $C0$ and 6 records belonging to class $C1$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	**p(j|t)**
Parent node	C0	10/20 = 0.5
Parent node	C1	10/20 = 0.5
Child node (M)	C0	6/10 = 0.6
Child node (M)	C1	4/10 = 0.4
Child node (F)	C0	4/10 = 0.4
Child node (F)	C1	6/10 = 0.6

The Gini Index is calculated as:

$$GINI(t) = 1 - \sum_{j=1}^{n} [p(j|t)]^2$$

The contingency tables for the parent and the two child nodes are as follows:

(a) Parent Node:

Class	Parent
C0	10
C1	10
	Gini (Parent) $= 1 - (0.5)^2 - (0.5)^2$ $= 1 - 0.5$ $= 0.5$

(b) Child Node for gender category M:

Class	$Gender = M$
C0	6
C1	4
	Gini (Child M) $= 1 - (0.6)^2 - (0.4)^2$ $= 1 - 0.52$ $= 0.48$

(c) Child Node for gender category F:

Class	$Gender = F$
C0	4
C1	6
	Gini (Child F) $= 1 - (0.4)^2 - (0.6)^2$ $= 1 - 0.52$ $= 0.48$

The gain for this split is calculated as:

$$\begin{aligned} Gain &= Gini(Parent) - [10/20 * (Gini(ChildM)) + 10/20 * (Gini(ChildF))] \\ &= 0.5 - [1/2 * (0.96)] \\ &= 0.5 - 0.48 \\ &= 0.02 \end{aligned}$$

(ii) Car Type - On splitting we get three child nodes corresponding to *Family*, *Sports* and *Luxury* categories respectively, with 4, 8 and 8 records in the three child nodes. For *Family* category, there is 1 record belonging to class $C0$ and 3 records belonging to class $C1$. For *Sports* category, there are 8 records belonging to class $C0$ and 0 records belonging to class $C1$. For *Luxury* category, there is 1 records belonging to class $C0$ and 7 records belonging to class $C1$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	**p(j|t)**
Parent node	C0	$10/20 = 0.5$
Parent node	C1	$10/20 = 0.5$
Child node ($Family$)	C0	$1/4 = 0.25$
Child node ($Family$)	C1	$3/4 = 0.75$
Child node ($Sports$)	C0	$8/8 = 1$
Child node ($Sports$)	C1	$0/8 = 0$
Child node ($Luxury$)	C0	$1/8 = 0.125$
Child node ($Luxury$)	C1	$7/8 = 0.875$

The contingency tables for the parent and the two child nodes are as follows:

(a) Parent Node:

Class	**Parent**
C0	10
C1	10
	Gini (Parent) $= 1 - (0.5)^2 - (0.5)^2$ $= 1 - 0.5$ $= 0.5$

(b) Child Node for Car Type $Family$:

Class	$CarType = Family$
C0	1
C1	3
	Gini ($Family$) $= 1 - (0.25)^2 - (0.75)^2$ $= 1 - 0.625$ $= 0.375$

(c) Child Node for Car Type $Sports$:

Class	$CarType = Sports$
C0	8
C1	0
	Gini ($Sports$) $= 1 - (1)^2 - (0)^2$ $= 1 - 1$ $= 0$

(d) Child Node for Car Type $Luxury$:

Class	$CarType = Luxury$
C0	1
C1	7
	Gini ($Luxury$) $= 1 - (0.125)^2 - (0.875)^2$ $= 1 - 0.781$ $= 0.219$

The gain for this split is calculated as:

$$\begin{aligned} Gain = Gini(Parent) - [4/20 * (Gini(Child_{Family})) \\ + 8/20 * (Gini(Child_{Sports})) + 8/20 * (Gini(Child_{Luxury}))] \end{aligned}$$

$$\begin{aligned} &= 0.5 - [4/20 * (0.375) + 8/20 * (0) + 8/20 * (0.219)] \\ &= 0.5 - 0.1626 \\ &= 0.3374 \end{aligned}$$

---

[20] 5.

(a) The Entropy computations for parent and child nodes when split is made on A and B attributes are as follows:

(i) Splitting on attribute A, we get two child nodes corresponding to T and F categories, with 7 and 3 records respectively. For T category, there are 4 records belonging to class $+$ and 3 records belonging to class $-$. For F category, there are 0 records belonging to class $+$ and 3 records belonging to class $-$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	**p(j|t)**
Parent node	$+$	$4/10 = 0.4$
Parent node	$-$	$6/10 = 0.6$
Child node (T)	$+$	$4/7 = 0.571$
Child node (T)	$-$	$3/7 = 0.428$
Child node (F)	$+$	$0/3 = 0$
Child node (F)	$-$	$3/3 = 1$

The Entropy for a node t is calculated as:

$$Entropy(t) = -\sum_{j=1}^{n} p(j|t) \log_2 p(j|t)$$

The contingency tables for the parent and the two child nodes are as follows:

(a) Parent Node:

Class	Parent
$+$	4
$-$	6
	$Entropy(Parent) = -[(0.4)\log_2(0.4) + (0.6)\log_2(0.6)]$ $= -[-0.528 - 0.442]$ $= 0.97$

(b) Child Node for attribute A category T:

Class	$A = T$
$+$	4
$-$	6
	$Entropy(Child_T) = -[(0.571)\log_2(0.571) + (0.428)\log_2(0.428)]$ $= -[-0.461 - 0.524]$ $= 0.985$

(c) Child Node for attribute A category F:

Class	$A = F$
$+$	0
$-$	3
	$Entropy(Child_F) = -[0\log_2(0) + 1\log_2(1)]$ $= -[-0 - 0]$ $= 0$

The gain for this split is calculated as:

$$\begin{aligned} Gain &= Entropy(Parent) - [7/10 * (Entropy(Child_T)) + 3/10 * (Entropy(Child_F))] \\ &= 0.97 - [0.7 * 0.985 + 0.3 * 0] \\ &= 0.97 - 0.689 \\ &= 0.280 \end{aligned}$$

(ii) Splitting on attribute B, we get two child nodes corresponding to T and F categories, with 4 and 6 records respectively. For T category, there are 3 records belonging to class $+$ and 1 record belonging to class $-$. For F category, there is 1 record belonging to class $+$ and 5 records belonging to class $-$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	$\mathbf{p(j|t)}$
Parent node	$+$	$4/10 = 0.4$
Parent node	$-$	$6/10 = 0.6$
Child node (T)	$+$	$3/4 = 0.75$
Child node (T)	$-$	$1/4 = 0.25$
Child node (F)	$+$	$1/6 = 0.166$
Child node (F)	$-$	$5/6 = 0.833$

The contingency tables for the parent and the two child nodes are as follows:

(a) Parent Node:

Class	Parent
$+$	4
$-$	6
	$Entropy(Parent) = -[(0.4)\log_2(0.4) + (0.6)\log_2(0.6)]$ $= -[-0.528 - 0.442]$ $= 0.97$

(b) Child Node for attribute B category T:

Class	$B = T$
$+$	3
$-$	1
	$Entropy(Child_T) = -[(0.75)\log_2(0.75) + (0.25)\log_2(0.25)]$ $= -[-0.311 - 0.5]$ $= 0.811$

(c) Child Node for attribute B category F:

Class	$B = F$
$+$	1
$-$	5
	$Entropy(Child_F) = -[0.166\log_2(0.166) + 0.833\log_2(0.833)]$ $= -[-0.430 - 0.219]$ $= 0.649$

The gain for this split is calculated as:

$$\begin{aligned} Gain &= Entropy(Parent) - [4/10 * (Entropy(Child_T)) + 6/10 * (Entropy(Child_F))] \\ &= 0.97 - [0.4 * 0.811 + 0.6 * 0.649] \\ &= 0.97 - 0.7138 \\ &= 0.2562 \end{aligned}$$

The decision tree induction algorithm would choose attribute A over B for splitting the tree. This is because splitting on attribute A results in a larger *Gain* (=0.280) as compared to the gain resulting from splitting of attribute B (=0.256).

---

[20] 6.

---

(1) The Classification Error computations for parent and child nodes when split is made on A and B attributes are as follows:

(a) Splitting on attribute A, we get two child nodes corresponding to T and F categories, with 50 records in each node. For T category, there are 20 records belonging to class $+$ and 30 records belonging to class $-$. For F category, there are 15 records belonging to class $+$ and 35 records belonging to class $-$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	**p(j|t)**
Parent node	$+$	$35/100 = 0.35$
Parent node	$-$	$65/100 = 0.65$
Child node (T)	$+$	$20/50 = 0.4$
Child node (T)	$-$	$30/50 = 0.6$
Child node (F)	$+$	$15/50 = 0.3$
Child node (F)	$-$	$35/50 = 0.7$

The Classification Error for a node t is calculated as:

$$Error(t) = 1 - \max_{j=1}^{n}[p(j|t)]$$

The following table gives the Classification Error for the parent and the two child nodes when splitting by attribute A:

Node	Error Computation
$Parent$	$Error = 1 - \max(0.35, 0.65)$ $= 0.35$
$Child_T$	$Error = 1 - \max(0.4, 0.6)$ $= 0.4$
$Child_F$	$Error = 1 - \max(0.3, 0.7)$ $= 0.3$

The gain for this split is calculated as:

$$\begin{aligned}
Gain &= Error(Parent) - [50/100 * (Error(Child_T)) + 50/100 * (Error(Child_F))] \\
&= 0.35 - [0.5 * 0.4 + 0.5 * 0.3] \\
&= 0
\end{aligned}$$

(b) Splitting on attribute B, we get two child nodes corresponding to T and F categories, with 35 and 65 records respectively. For T category, there are 15 records belonging to class $+$ and 20 records belonging to class $-$. For F category, there are 20 records belonging to class $+$ and 45 records belonging to class $-$. The relative frequencies ($p(j|t)$) for each class j on each node t are:

t	**Class**	**p(j|t)**
Parent node	$+$	$35/100 = 0.35$
Parent node	$-$	$65/100 = 0.65$
Child node (T)	$+$	$15/35 = 0.428$
Child node (T)	$-$	$20/35 = 0.571$
Child node (F)	$+$	$20/65 = 0.307$
Child node (F)	$-$	$45/65 = 0.692$

The following table gives the Classification Error for the parent and the two child nodes when splitting by attribute B:

Node	Error Computation
$Parent$	$Error = 1 - \max(0.35, 0.65)$ $= 0.35$
$Child_T$	$Error = 1 - \max(0.428, 0.571)$ $= 0.428$
$Child_F$	$Error = 1 - \max(0.307, 0.692)$ $= 0.307$

The gain for this split is calculated as:

$$\begin{aligned}
Gain &= Error(Parent) - [35/100 * (Error(Child_T)) + 65/100 * (Error(Child_F))] \\
&= 0.35 - [0.35 * 0.428 + 0.65 * 0.307] \\
&= 0
\end{aligned}$$

Since, we obtain the same gain of 0 by splitting on either A or B (as computed in (a) and (b) above), any of the two attributes (A or B) can be chosen as the first splitting attribute.

(2) When we split by attribute A, we obtain 20 and 30 records corresponding the + and − classes respectively, for the T category. Also, we obtain 15 and 35 records corresponding the + and − classes respectively, for the F category. Similarly, when we split by attribute B, we obtain 15 and 20 records corresponding the + and − classes respectively, for the T category. Also, we obtain 20 and 45 records corresponding the + and − classes respectively, for the F category. We can use this information along with the cost matrix to compute the cost associated with each split. The following table gives the cost computations for splitting on attributes A and B:

Attribute	**Total cost of split**
A	$Cost = (20*(-1)+30*(0))+(15*(100)+35*(-10))$ $= 1130$
B	$Cost = (15*(-1)+20*(0))+(20*(100)+45*(-10))$ $= 1535$

Since, splitting by attribute A gives a lower cost (1130) as compared to splitting on attribute B (1535), A should be chosen as the first splitting attribute.