I Neha Moolchandani declare that I have completed this assignment completely and entirely on my own, without any consultation with others. I understand that any breach of the UAB Academic Honor Code may result in severe penalties.

Chap 2:

PART I: Exercises 2.6 (20 pts), 2.8 (50 pts (30 pts for (a) and 20 pts for (b).) For the normalization step in (b), you should normalize each value by dividing it by the length (Euclidean norm) of that data point. Do not forget to normalize the query point as well.

2.6 Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the *Euclidean distance* between the two objects.

(b) Compute the *Manhattan distance* between the two objects.

(c) Compute the *Minkowski distance* between the two objects, using $q = 3$.

(d) Compute the *supremum distance* between the two objects.

a) d = √[(p1 – q1)^2 + (p2 – q2)^2].

d = √[(22-20)^2 + (1-0)^2 + (42-36)^2 + (10-8)^2]

d= √[(2)^2 + (1)^2 + (6)^2 + (2)^2]

d=√45 => 6.7082

b) d = |p1 − q2| + |p1 − q2|

d= |22-20| + |1-0| + |42-36| +|10-8| = 11

c)

$$d(i, j) = \sqrt[p]{|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{il} - x_{jl}|^p}$$

d= 3√[|22-20|^3 + |1-0|^3 + |42-36|^3 + |10-8|^3]

d=√233 = 6.15434

d)

$$d(i,j) = \lim_{p\to\infty} \sqrt[p]{|x_{i1}-x_{j1}|^p + |x_{i2}-x_{j2}|^p + \cdots + |x_{il}-x_{jl}|^p} = \max_{f=1}^{l} |x_{if}-x_{jf}|$$

d=|42-36| => 6

2.8 It is important to define or select similarity measures in data analysis. However, there is no commonly accepted subjective similarity measure. Results can vary depending on the similarity measures used. Nonetheless, seemingly different similarity measures may be equivalent after some transformation.

Suppose we have the following 2-D data set:

	A_1	A_2
x_1	1.5	1.7
x_2	2	1.9
x_3	1.6	1.8
x_4	1.2	1.5
x_5	1.5	1.0

(a) Consider the data as 2-D data points. Given a new data point, $\boldsymbol{x} = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using Euclidean distance, Manhattan distance, supremum distance, and cosine similarity.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.

a)

Euclidean:

d = √[(p1 – q1 1)^2 + (p2 – q2)^2].

X1: d = √[(1.4-1.5)^2 + (1.6-1.7)^2] = 0.141

X2: d = √[(1.4-2)^2 + (1.6-1.9)^2 = 0.671

X3: d = √[(1.4-1.6)^2 + (1.6-1.6)^2 = 0.283

X4: d = √[(1.4-1.2)^2 + (1.6-1.5)^2 = 0.224

X5: d = √[(1.4-1.5)^2 + (1.6-1.0)^2 = 0.608

Ranking based on **Euclidean**: x1,x4,x3,x5,x2

Manhattan:

d = |p1 − q2| + |p1 − q2|

X1: d= |1.4-1.5| + |1.6-1.7| = 0.2

X2: d= |1.4-2| + |1.6-1.9| = 0.9

X3: d= |1.4-1.6| + |1.6-1.8| = 0.4

X4: d= |1.4-1.2| + |1.6-1.5| = 0.3

X5: d= |1.4-1.5| + |1.6-1.0| = 0.7

Ranking based on **Manhattan**: x1, x4,x3,x5,x2

Supremum:

X1: d = |1.4-1.5| or |1.6-1.7| => 0.1

X2: d = |1.4-2| or |1.6-1.9| => 0.6

X3: d = |1.4-1.6| or |1.6-1.8| => 0.2

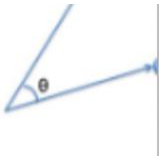
X4: d = |1.4-1.2| or |1.6-1.5| => 0.2

X5: d = |1.4-1.5| or |1.6-1.0| => 0.6

Ranking based on **Supremum**: x1, x4,x3,x5,x2

Cosine Similarity:

Calculating Cosine Similarity: $cos(d_1, d_2) = \frac{d_1 \bullet d_2}{\| d_1 \| \times \| d_2 \|}$ $sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$



where • indicates vector dot product, ||*d*|| : the length of vector *d*

x: (1.4,1.6)

y: (1.5, 1.7) => (1.4x1.5 + 1.6x1.9) / (√[((1.4)^2+(1.6)^2 x √[((1.5^2)+(1.7^2))

⇨ 4.189/4.791 => 0.99

x: (1.4,1.6)

y: (2.0, 1.9) => (1.4x2 + 1.6x1.7) / (√[((1.4)^2+(1.6)^2 x √[((2.0^2)+(1.9^2))

⇨ 5.84/5.89 => 0.9849

x: (1.4,1.6)

y: (1.6, 1.8) => (1.4x1.6 + 1.6x1.8) / (√[((1.4)^2+(1.6)^2 x √[((1.6^2)+(1.8^2))

⇨ 5.12056/5.12001 => 0.9999

x: (1.4,1.6)

y: (1.2, 1.5) => (1.4x1.6 + 1.2x1.5) / (√[((1.4)^2+(1.6)^2 x √[((1.2^2)+(1.5^2))

⇨ 4.04/4.08396 => 0.992

x: (1.4,1.6)

y: (1.5, 1.0) => (1.4x1.6 + 1.5x1.0) / (√[((1.4)^2+(1.6)^2 x √[((1.5^2)+(1.0^2))

⇨ 3.74/3.83275 => 0.975799

Ranking based on **Cosine Similarity**: x1, x4,x3,x2,x5

b) Normalizing the data to make norm of each data equal to 1.

Normalization: Map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Range of Values: Lengh of a query point (1.4,1.6) use squareroot(x1sqr+x2sqr) , divide the resulting length with each point to get normalized form of that point.

Length of Query Point: (1.4,1.6) = √[1.4^2 + 1.6^2] = 2.126

	A1	A2

X1	1.5/2.23 = 0.6726	1.7/2.23 = 0.76233
X2	2/2.193 = 0.9119	1.9/2.193 = 0.6884
X3	1.6/2.41 = 0.6639	1.8/2.41 = 0.7469
X4	1.2/1.92 = 0.625	1.5/1.92 = 0.78125
X5	1.5/1.8027 = 0.8321	1/1.92 = 0.5208
X(1.4,1.6)	1.4/2.126 = 0.6585	1.6/1.8027 = 0.7526

Using Euclidean Distance: Respect to Query Point: (1.4,1.6)

d = √[(p1 – q1)^2 + (p2 – q2)^2].

X1: d =√[(0. 6639-0. 6726)^2 + (0. 76233-0. 7526)^2] = 0.0786

X2: d =√[(0. 9119-0.6585)^2 + (0. 6884-0. 7526)^2] = 0.0299

X3: d =√[(0. 6639-0.6585)^2 + (0. 8664-0. 7526)^2] = 0.0818

X4: d = √[(0. 625-0.6585)^2 + (0. 7469-0.7526)^2] = 0.1487

X5: d = √[(0. 8321-0.6585)^2 + (0. 78125-0.7526)^2] = 0.26286

Ranking: x2, x3, x1, x4, x5

PART II: For the same data from 2.8, do z-score normalization (see Lecture 2: Mean Absolute Deviation approach) for each feature dimension (A1 and A2, respectively). Please include the new point *x*(1.4, 1.6) in the calculation of mean and mean absolute deviation. Use Euclidean distance on the normalized data to rank the data points in the ascending order of their distance values. (10 pts for getting the z-scores correctly, and 10 points for getting the rank correctly.)

$$MAD = \frac{\sum |x - \bar{X}|}{n}$$

Mean: (1.0+ 1.2+1.5+1.5+1.5+1.6+1.7+1.8+1.9+2.0+1.4+1.6) / 12 = 1.55

Finding all distances per point:

((1.5 - 1.53) + (2 - 1.53) + (1.6 - 1.53) + (1.2 - 1.53) + (1.5 - 1.53) + (1.4 - 1.53) +

A2: ((1.0 - 1.583) + (1.5 - 1.583) + (1.7 - 1.583) + (1.8 - 1.583) + (1.9 - 1.583) + (1.6 - 1.583)) / 6

Add all Distances up: MAD

MAD: A1: 0.1778

Mean A1: 1.533

MAD A2: 0.22

Mean A2: 1.583

(MEAN,MAD) and (A1,A2)

Z-Score: Score-Mean/Deviation

	A1	A2
X1	(1.5-1.53)/0.1778 = -0.169	(1.7-1.58)/0.22 = 0.5454
X2	(2-1.53)/ 0.1778 = -2.64	(1.9-1.58)/0.22 = 1.4545
X3	(1.6-1.53)/ 0.1778 = 0.3937	(1.8-1.58)/0.22 = 1.0
X4	(1.2-1.53)/ 0.1778 = -1.856	(1.5-1.58)/0.22 = -0.3636
X5	(1.5-1.53)/ 0.1778 = -0.169	(1.0-1.58)/0.22 = -2.636

MAD: A1: 0.1778

Mean A1: 1.533

MAD A2: 0.22

Mean A2: 1.583

(MEAN,MAD) and (A1,A2)

MAD Together: 0.2083

MEAN Together: 1.558

(MEAN,MAD) (A1,A2)

Using Euclidean Distance: Respect to Query Point: (1.4,1.6)

d = √[(p1 – q1)^2 + (p2 – q2)^2].

X1: d =√[(1.558+0. 169)^2 + (0. 2083-0. 5454)^2/] = 1.759

X2: d =√[(1.558+2.64)^2 + (0. 2083-0. 4545)^2/] = 4.207

X3: d =√[(1.558-0.3937)^2 + (0. 2083-1)^2] = 1.4099

X4: d = √[(1.558+1.859)^2 + (0. 2083+0. 3636)^2] = 3.4646

X5: d = √[(1.558 +0. 169)^2 + (0. 2083+2.636)^2] = 3.328

Ranking: x3, x1, x5, x4, x2

PART III: Read Section 2.4.6 and calculate the distance between (X1, X2) and that between (X1, X3) according to the data in the table below. Please include the details of your calculation. (20 pts)

	A1 (NOMINAL)	A2 (NUMERIC)	A3 (ORDINAL)	A4 (Asymmetric Binary)	A5 (Asymmetric Binary)

X1	A	100	Small	0	1
X2	B	20	Medium	1	0
X3	C	50	Large	1	1

(X1, X2) :
Nominal: Because A doesn't equal B => 0 for AB
Numeric: |100-20| / (100-20) => 1
Ordinal: Small and Medium => Medium

(X1, X3)
Nominal: Because A doesn't equal C =>0 for AC
Numeric: |100-50| / (100-20) => 50/80 =>0.626
Ordinal: Small and Large => Medium

2.4.6 Dissimilarity for Attributes of Mixed Types

Sections 2.4.2 through 2.4.5 discussed how to compute the dissimilarity between objects described by attributes of the same type, where these types may be either *nominal*, *symmetric binary*, *asymmetric binary*, *numeric*, or *ordinal*. However, in many real databases, objects are described by a *mixture* of attribute types. In general, a database can contain all of these attribute types.

"*So, how can we compute the dissimilarity between objects of mixed attribute types?*" One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.

A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0, 1.0].

Suppose that the data set contains p attributes of mixed type. The dissimilarity $d(i, j)$ between objects i and j is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}, \tag{2.22}$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of attribute f for object i or object j), or (2) $x_{if} = x_{jf} = 0$ and attribute f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of attribute f to the dissimilarity between i and j (i.e., $d_{ij}^{(f)}$) is computed dependent on its type:

- If f is numeric: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, where h runs over all nonmissing objects for attribute f.
- If f is nominal or binary: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise, $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

These steps are identical to what we have already seen for each of the individual attribute types. The only difference is for numeric attributes, where we normalize so that the values map to the interval [0.0, 1.0]. Thus, the dissimilarity between objects can be computed even when the attributes describing the objects are of different types.