1

Introduction

1. [Fall 2008]

   For each data set given below, give specific examples of classification, clustering, association rule mining, and anomaly detection tasks that can be performed on the data. For each task, state how the data matrix should be constructed (i.e., specify the rows and columns of the matrix).

   (a) Ambulatory Medical Care data[1], which contains the demographic and medical visit information for each patient (e.g., gender, age, duration of visit, physician's diagnosis, symptoms, medication, etc).

   Answer:

Classification
Task: Diagnose whether a patient has a disease. Row: Patient Column: Patient's demographic and hospital visit information (e.g., symptoms), along with a class attribute that indicates whether the patient has the disease.
Clustering
Task: Find groups of patients with similar medical conditions Row: A patient visit Column: List of medical conditions of each patient
Association rule mining
Task: Identify the symptoms and medical conditions that co-occur together frequently Row: A patient visit Column: List of symptoms and diagnosed medical conditions of the patient
Anomaly detection
Task: Identify healthy looking patients with rare medical disorders Row: A patient visit Column: List of demographic attributes, symptoms, and medical test results of the patient

[1] See for example, the National Hospital Ambulatory Medical Care Survey `http://www.cdc.gov/nchs/about/major/ahcd/ahcd1.htm`

(b) Stock market data, which include the prices and volumes of various stocks on different trading days.

Answer:

Classification
Task: Predict whether the stock price will go up or down the next trading day Row: A trading day Column: Trading volume and closing price of the stock the previous 5 days and a class attribute that indicates whether the stock went up or down
Clustering
Task: Identify groups of stocks with similar price fluctuations Row: A company's stock Column: Changes in the daily closing price of the stock over the past ten years
Association rule mining
Task: Identify stocks with similar fluctuation patterns(e.g., {Google-Up, Yahoo-Up}) Row: A trading day Column: List of all stock-up and stock-down events on the given day.
Anomaly detection
Task: Identify unusual trading days for a given stock (e.g., unusually high volume) Row: A trading day Column: Trading volume, change in daily stock price (daily high − low prices), and average price change of its competitor stocks

(c) Database of Major League Baseball (MLB).

Classification
Task: Predict the winner of a game between two MLB teams. Row: A game. Column: Statistics of the home and visiting teams over their past 10 games they had played (e.g., average winning percentage and hitting percentage of their players)
Clustering
Task: Identify groups of players with similar statistics Row: A player Column: Statistics of the player
Association rule mining
Task: Identify interesting player statistics (e.g., 40% of right-handed players have a batting percentage below 20% when facing left-handed pitchers) Row: A player Column: Discretized statistics of the player
Anomaly detection
Task: Identify players who performed considerably better than expected in a given season Row: A (player,season) pair e.g, (player1 in 2007) Column: Ratio statistics of a player (e.g., ratio of average batting percentage in 2007 to career average batting percentage)

2

Data

2.1 Types of Attributes

1. Classify the following attributes as binary, discrete, or continuous. Also classify them as qualitative (nominal or ordinal) or quantitative (interval or ratio). Some cases may have more than one interpretation, so briefly indicate your reasoning if you think there may be some ambiguity.

   (a) Number of courses registered by a student in a given semester.
   Answer: Discrete, quantitative, ratio.

   (b) Speed of a car (in miles per hour).
   Answer: Discrete, quantitative, ratio.

   (c) Decibel as a measure of sound intensity.
   Answer: Continuous, quantitative, interval or ratio. It is actually a logratio type (which is somewhere between interval and ratio).

   (d) Hurricane intensity according to the Saffir-Simpson Hurricane Scale.
   Answer: Discrete, qualitative, ordinal.

   (e) Social security number.
   Answer: Discrete, qualitative, nominal.

2. Classify the following attributes as:

   - discrete or continuous.
   - qualitative or quantitative
   - nominal, ordinal, interval, or ratio

Some cases may have more than one interpretation, so briefly indicate your reasoning if you think there may be some ambiguity.

(a) Julian Date, which is the number of days elapsed since 12 noon Greenwich Mean Time of January 1, 4713 BC.
Answer: Continuous, quantitative, interval

(b) Movie ratings provided by users (1-star, 2-star, 3-star, or 4-star).
Answer: Discrete, qualitative, ordinal

(c) Mood level of a blogger (cheerful, calm, relaxed, bored, sad, angry or frustrated).
Answer: Discrete, qualitative, nominal

(d) Average number of hours a user spent on the Internet in a week.
Answer: Continuous, quantitative, ratio

(e) IP address of a machine.
Answer: Discrete, qualitative, nominal

(f) Richter scale (in terms of energy release during an earthquake).
Answer: Continuous, qualitative, ordinal
In terms of energy release, the difference between 0.0 and 1.0 is not the same as between 1.0 and 2.0. Ordinal attributes are qualitative; yet, can be continuous.

(g) Salary above the median salary of all employees in an organization.
Answer: Continuous, quantitative, interval

(h) Undergraduate level (freshman, sophomore, junior, and senior) for measuring years in college.
Answer: Discrete, qualitative, ordinal

3. For each attribute given, classify its type as:

- discrete or continuous AND
- qualitative or quantitative AND
- nominal, ordinal, interval, or ratio

Indicate your reasoning if you think there may be some ambiguity in some cases.

Example: Age in years.

Answer: Discrete, quantitative, ratio.

(a) Daily user traffic volume at YouTube.com (i.e., number of daily visitors who visited the Web site).
Answer: Discrete, quantitative, ratio.

(b) Air pressure of a car/bicycle tire (in psi).
Answer: Continuous, quantitative, ratio.

(c) Homeland Security Advisory System ratings - code red/orange/etc.
Answer: Discrete, qualitative, ordinal.

(d) Amount of seismic energy release, measured in Richter scale.
Answer: Continuous, qualitative, ordinal.

(e) Credit card number.
Answer: Discrete, qualitative, nominal.

(f) The wealth of a nation measured in terms of gross domestic product (GDP) per capita above the world's average of $10,500.
Answer: Continuous, quantitative, interval.

4. For each attribute given, classify its type as:

   - discrete or continuous AND
   - qualitative or quantitative AND
   - nominal, ordinal, interval, or ratio

   Indicate your reasoning if you think there may be some ambiguity in some cases.

   Example: Age in years.

   Answer: Discrete, quantitative, ratio.

   (a) Favorite movie of each person.
   Answer: Discrete, qualitative, nominal

   (b) Number of days since Jan 1, 2011.
   Answer: Discrete, quantitative, interval.

   (c) Category of a hurricane (The Saffir-Simpson Hurricane Wind Scale ranges from category 1 to category 5).
   Answer: Discrete, qualitative, ordinal.

   (d) Number of students enrolled in a class.
   Answer: Discrete, quantitative, ratio

5. For each attribute given, classify its type as:

   - discrete or continuous AND
   - qualitative or quantitative AND
   - nominal, ordinal, interval, or ratio

   Indicate your reasoning if you think there may be some ambiguity in some cases.

   Example: Temperature in Kelvin

   Answer: Continuous, quantitative, ratio.

   (a) Number of years since 1 BC. For example, 2 BC is year -1, 1 BC is year 0, 1 AD is year 1, and 2013 AD is year 2013 (note, there is no 0 AD in Gregorian calendar).

   Answer: Discrete/Continuous, quantitative, interval.

   (b) GPA of a student.

   Answer: Continuous, qualitative, ordinal.

   (c) Mood level of a blogger (cheerful, calm, relaxed, bored, sad, angry or frustrated).

   Answer: Discrete, qualitative, nominal.

   (d) Sound intensity in decibel scale.

   Answer: Continuous, qualitative, ordinal. In terms of sound intensity, the difference between 0dB and 1dB is not the same as the difference between 10 dB and 11 dB (decibels are in log scale); thus, it is not an interval attribute.

6. State the type of each attribute given below before and after we have performed the following transformation.

   (a) Hair color of a person is mapped to the following values: black = 0, brown = 1, red = 2, blonde = 3, grey = 4, white = 5.

   Answer: Nominal (both before and after transformation).

   (b) Grade of a student (from 0 to 100) is mapped to the following scale: A = 4.0, A- = 3.5, B = 3.0, B- = 2.5, C = 2.0, C- = 1.5, D = 1.0, D- = 0.5, E = 0.0

   Answer: Ratio (before transformation) to ordinal (after transformation).

(c) Age of a person is discretized to the following scale: Age < 12, $12 \leq$ Age < 21, $21 \leq$ Age < 45, $45 \leq$ Age < 65, Age > 65.

Answer: Ratio (before transformation) to ordinal (after transformation)

(d) Annual income of a person is discretized to the following scale: Income $<$ \$20K, \$20K $\leq$ Income $<$ \$60K, \$60K $\leq$ Income $<$ \$120K, \$120K $\leq$ Age $<$ \$250K, Age $\geq$ \$250K.

Answer: Ratio (before transformation) to ordinal (after transformation).

(e) Height of a person is changed from meters to feet.

Answer: Ratio (both before and after transformation)

(f) Height of a person is changed from meters to {Short, Medium, Tall}.

Answer: Ratio (before transformation) to ordinal (after transformation).

(g) Height of a person is changed from feet to number of inches above 4 feet.

Answer: Ratio (before transformation) to interval (after transformation).

(h) Weight of a person is standardized by subtracting it with the mean of the weight for all people and dividing by its standard deviation.

Answer: Ratio (before transformation) to interval (after transformation)

7. State whether it is meaningful (based on the properties of the attribute values) to apply the following operations to the data given below

(a) Average amplitude of seismic waves (in Richter scale) for the 10 deadliest earthquakes in Asia.

Answer: No because Richter scale is ordinal.

(b) Average number of characters in a collection of spam messages.

Answer: Yes because number of characters is a ratio attribute.

(c) Pearson's correlation between shirt size and height of an individual.

Answer: No because shirt size is ordinal.

(d) Median zipcode of households in the United States.

Answer: No because zipcode is nominal.

(e) Entropy of students (based on the GPA they obtained for a given course).

Answer: Yes because entropy is applicable to nominal attributes.

(f) Geometric mean of temperature (in Fahrenheit) for a given city.

Answer: No because temperature (in Fahrenheit) is not a ratio attribute.

2.2 Data Preprocessing

1. Consider the following dataset that contains the age and gender information for 9 users who visited a given website.

UserID	1	2	3	4	5	6	7	8	9
Age	17	24	25	28	32	38	39	49	68
Gender	Female	Male	Male	Male	Female	Female	Female	Male	Male

(a) Suppose you apply equal interval width approach to discretize the Age attribute into 3 bins. Show the userIDs assigned to each of the 3 bins.

Answer: Bin width $= \frac{68-17}{3} = \frac{51}{3} = 17$.

Bin 1: 1, 2, 3, 4, 5

Bin 2: 6, 7, 8

Bin 3: 9

(b) Repeat the previous question using the equal frequency approach.

Answer: Since there are 9 users and 3 bins, every bin must contain 3 users.

Bin 1: 1, 2, 3

Bin 2: 4, 5, 6

Bin 3: 7, 8, 9

(c) Repeat question (a) using a supervised discretization approach (with Gender as class attribute). Specifically, choose the bins in such a way that their members are as "pure" as possible (i.e., belonging to the same class).

Answer:

Bin 1: 1, 2, 3, 4

Bin 2: 5, 6, 7

Bin 3: 8, 9

2. Consider an attribute X of a data set that takes the values $\{x_1, x_2, \cdots, x_9\}$ (sorted in increasing order of magnitude). We apply two methods (equal interval width and equal frequency) to discretize the attribute into 3 bins. The bins obtained are shown below:

   Equal Width: $\{x_1, x_2, x_3\}$, $\{x_4, x_5, x_6, x_7, x_8\}$, $\{x_9\}$

   Equal Frequency: $\{x_1, x_2, x_3\}$, $\{x_4, x_5, x_6\}$, $\{x_7, x_8, x_9\}$

   Explain what will be the effect of applying the following transformations on each discretization method, i.e., whether the elements assigned to each bin can change if you discretize the attribute **after** applying the transformation function below. Note that $\bar{X}$ denotes the average value and σ_x denotes standard deviation of attribute X.

   (a) $\mathrm{X} \rightarrow \mathrm{X} - \bar{X}$ (i.e., if the attribute values are centered).

   Answer: No change for equal width because the distance between x_i and x_{i+1} is unchanged. No change for equal frequency because the relative ordering of data points remain the same (i.e., if $x_i < x_{i+1}$ then $x_i - \bar{X} < x_{i+1} - \bar{X}$).

   (b) $\mathrm{X} \rightarrow \frac{X-\bar{X}}{\sigma_x}$ (i.e., if the attribute values are standardized).

   Answer: Since the distances between every pair of points (x_i, x_{i+1}) change uniformly (by a constant factor of σ_x, the elements in the bins are unchanged for equal width discretization. No change for equal frequency because the relative ordering of data points remain the same.

   (c) $\mathrm{X} \rightarrow \exp\left[\frac{X-\bar{X}}{\sigma_x}\right]$ (i.e., if the values are standardized and exponentiated).

   Answer: The bin elements may change for equal width because the distances between x_i and x_{i+1} may not change uniformly. No change for equal frequency because the relative ordering of data points remain the same.

3. Consider a dataset that has 3 attributes (x_1, x_2, and x_3). The distribution of each attribute is as follows and shown in Figure

   - x_1 has a uniform distribution in the range between 0 and 1.
   - x_2 is generated from a mixture of 3 Gaussian distributions centered at 0.1, 0.5, and 0.9, respectively. The standard deviation of the

distributions are 0.02, 0.1, and 0.02, respectively. Assume each point is generated from one of the 3 distributions and the number of points associated with each distribution is different.

- x_3 is generated from an exponential distribution with mean 0.1.

(a) Which attribute(s) is likely to produce the same bins regardless of whether you use equal width or equal frequency approaches (assuming the number of bins is not too large).

Answer: x_1.

(b) Which attribute(s) is more suitable for equal frequency than equal width discretization approaches.

Answer: x_3.

(c) Which attribute(s) is not appropriate for both equal width and equal frequency discretization approaches.

Answer: x_2.

(d) If all 3 are initially ratio attributes, what are their attribute types after discretization?

Answer: Ordinal.

4. An e-commerce company is interested in identifying the highest spending customers at its online store using association rule mining. One of the rules identified is:

$$21 \leq \text{Age} < 45 \text{ AND NumberOfVisits} > 50 \rightarrow \text{AmountSpent} > \$500,$$

where the Age attribute was discretized into 5 bins, NumberOfVisits was discretized into 8 bins, and AmountSpent was discretized into 8 bins. The confidence of an association rule $A, B \rightarrow C$ is defined as

$$\text{Confidence}(A, B \rightarrow C) = P(C|A, B) = \frac{P(A, B, C)}{P(A, B)} \tag{2.1}$$

where $P(C|A, B)$ is the conditional probability of C given A and B, $P(A, B, C)$ is the joint probability of A, B, and C, and $P(A, B)$ is the joint probability of A and B. The probabilities are empirically estimated based on their relative frequencies in the data. For example, $P(\text{AmountSpent} > \$500)$ is given by the proportion of online users who visited the store and spent more than \$500.

(a) Suppose we increase the number of bins for the Age attribute from 5 to 6 so that the discretized Age in the rule becomes $21 \leq \text{Age} < 30$ instead of $21 \leq \text{Age} < 45$, will the confidence of the rule be non-increasing, non-decreasing, stays the same, or could go either way (increase/decrease)?

Answer: Can increase/decrease.

(b) Suppose we increase the number of bins for the AmountSpent attribute from 8 to 10, so that the right hand side of the rule becomes $\$500 < \text{AmountSpent} < \$1000$, will the confidence of the rule be non-increasing, non-decreasing, stays the same, or could go either way (increase/decrease)?

Answer: Non-increasing.

(c) Suppose the values for NumberOfVisits attribute are distributed according to a Poisson distribution with a mean value equals to 4. If we discretize the attribute into 4 bins using the equal frequency approach, what are the bin values after discretization? Hint: you need to refer to the cumulative distribution table for Poisson distribution to answer the question.

Answer: Choose the bin values such that the cumulative distribution is close to 0.25, 0.5, and 0.75. This corresponds to bin values: 0 to 2, 3, 4 to 5, and greater than 5.

5. Null values in data records may refer to missing or inapplicable values. Consider the following table of employees for a hypothetical organization:

Name	Sales commission	Occupation
John	5000	Sales
Mary	1000	Sales
Bob	null	Non-sales
Lisa	null	Non-sales

The null values in the table refer to inapplicable values since sales commission are calculated for sales employees only. Suppose we are interested to calculate the similarity between users based on their sales commission.

(a) Explain what is the limitation of the approach to compute similarity if we replace the null values in sales commission by 0.

Answer: Mary will be more similar to Bob and Lisa than to John.

(b) Explain what is the limitation of the approach to compute similarity if we replace the null values in sales commission by the average value of sales commission (i.e., 3000).

Answer: Both Mary and John are less similar to each other than to Bob and Lisa.

(c) Propose a method that can handle null values in the sales commission so that employees that have the same occupation are closer to each other than to employees that have different occupations.

Answer: One way is to change the similarity function as follows:

$$\text{Similarity}(a,b) = \begin{cases} \infty, & \text{if both a and b are null;} \\ 0, & \text{if one of a or b is null;} \\ s(a,b), & \text{otherwise.} \end{cases}$$

where $s(a,b)$ is the original similarity measure used for the sales commission.

6. Consider a data set from an online social media Web site that contains information about the age and number of friends for 5,000 users.

(a) Suppose the number of friends for each user is known. However, only 4000 out of 5000 users provide their age information. The average age of the 4,000 users is 30 years old. If you replace the missing values for age with the value 30, will the average age computed for the 5,000 users increases, decreases, or stays the same (as 30)?

Answer: Average age does not change.

$$\begin{aligned} \overline{x}_{\text{old}} &= \frac{1}{4000}\sum_{i=1}^{4000} x_i \\ \overline{x}_{\text{new}} &= \frac{1}{5000}\sum_{i=1}^{5000} x_i = \frac{1}{5000}\left[\sum_{i=1}^{4000} x_i + \sum_{i=4001}^{5000} x_i\right] \end{aligned}$$

Since $x_i = \overline{x}_{\text{old}}$ for $i = 4001, 4002, \cdots, 5000$ and $\sum_{i=1}^{4000} x_i = 4000\overline{x}_{\text{old}}$, we have

$$\overline{x}_{\text{new}} = \frac{1}{5000}\left[4000\overline{x}_{\text{old}} + 1000\overline{x}_{\text{old}}\right] = \overline{x}_{\text{old}}$$

(b) Suppose the covariance between age and number of friends calculated using the 4,000 users (with no missing values) is 20. If you replace the missing values for age with the average age of the 4,000 users, would the covariance between age and number of friends increases, decreases, or stays the same (as 20)? Assume that the average number of followers for all 5,000 users is the same as the average for 4,000 users.

Answer: Covariance will decrease. Let $C_1 = \sum_{i=1}^{4000}(x_i - \overline{x})(y_i - \overline{y})/3999$ be the covariance computed using the 4,000 users without missing values. If we impute the missing values for age with average age, $\overline{x}$ remains unchanged according to part (a). Furthermore, $\overline{y}$ is assumed to be unchanged. Thus, the new covariance is

$$\begin{aligned} C_2 &= \frac{1}{4999}\sum_{i=1}^{5000}(x_i - \overline{x})(y_i - \overline{y}) \\ &= \frac{1}{4999}\left[\sum_{i=1}^{4000}(x_i - \overline{x})(y_i - \overline{y}) + \sum_{i=4001}^{5000}(x_i - \overline{x})(y_i - \overline{y})\right] \\ &= \frac{1}{4999}\left[\sum_{i=1}^{4000}(x_i - \overline{x})(y_i - \overline{y}) + \sum_{i=4001}^{5000}(\overline{x} - \overline{x})(y_i - \overline{y})\right] \\ &= \frac{1}{4999}\sum_{i=1}^{4000}(x_i - \overline{x})(y_i - \overline{y}) < C_1 \end{aligned} \tag{2.2}$$

7. Consider the following data matrix on the right, in which two of its values are missing (the matrix on the left shows its true values).

$$\begin{bmatrix} -0.2326 & 0.2270 \\ -0.0847 & 0.7125 \\ 0.1275 & 0.3902 \\ 0.1329 & -0.1461 \\ 0.3724 & 0.1756 \\ 0.4975 & 0.8536 \\ 0.6926 & 0.7834 \\ 0.7933 & 0.7375 \\ 0.8229 & 0.2147 \\ 0.8497 & 0.4980 \\ 1.0592 & 0.7600 \\ 1.5028 & 1.0122 \end{bmatrix} \longrightarrow \begin{bmatrix} -0.2326 & 0.2270 \\ -0.0847 & 0.7125 \\ 0.1275 & 0.3902 \\ ? & -0.1461 \\ 0.3724 & 0.1756 \\ 0.4975 & 0.8536 \\ 0.6926 & 0.7834 \\ 0.7933 & 0.7375 \\ 0.8229 & 0.2147 \\ 0.8497 & 0.4980 \\ 1.0592 & ? \\ 1.5028 & 1.0122 \end{bmatrix}$$

(a) Impute the missing values for the matrix on the right by their respective column averages. Show the imputed values and calculate their root-mean-square-error (RMSE).

$$\text{RMSE} = \sqrt{\frac{(\mathbf{A}_{4,1} - \tilde{\mathbf{A}}_{4,1})^2 + (\mathbf{A}_{11,2} - \tilde{\mathbf{A}}_{11,2})^2}{2}}$$

where $\mathbf{A}_{i,j}$ denotes the true value of the (i, j)-th element of the data matrix and $\tilde{\mathbf{A}}_{i,j}$ denotes its corresponding imputed value.

Answer: The column averages are [0.5819 0.4962]. The imputed values are

$$\begin{bmatrix} -0.2326 & 0.2270 \\ -0.0847 & 0.7125 \\ 0.1275 & 0.3902 \\ 0.5819 & -0.1461 \\ 0.3724 & 0.1756 \\ 0.4975 & 0.8536 \\ 0.6926 & 0.7834 \\ 0.7933 & 0.7375 \\ 0.8229 & 0.2147 \\ 0.8497 & 0.4980 \\ 1.0592 & 0.4962 \\ 1.5028 & 1.0122 \end{bmatrix}$$

and the RMSE value is

$$\text{RMSE} = \sqrt{\frac{(0.1329 - 0.5819)^2 + (0.7600 - 0.4962)^2}{2}} = 0.3683$$

(b) The Expectation-Maximization (E-M) algorithm is a well-known approach for imputing missing values. Assuming the data is generated from a multivariate Gaussian distribution, E-M iteratively computes the following conditional mean for each attribute and uses it to impute the missing values:

$$\mu_{i|j} = \hat{\mu}_i + \mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{jj}^{-1}(\mathbf{x}_j - \hat{\mu}_j)$$

where the indices $i, j \in \{1, 2\}$ refer to one of the two attributes of the data and Σ^{-1} denote inverse of the covariance matrix. Repeat the previous question by applying the E-M algorithm iteratively for

5 times. Assume the covariance matrix of the data is known and given by

$$\mathbf{\Sigma} = \begin{bmatrix} 0.25 & 0.1 \\ 0.1 & 0.15 \end{bmatrix}$$

In the first iteration, compute the mean value for each column using only the non-missing values. In subsequent iterations, compute the mean value for each column using both the non-missing and imputed values. Show the imputed values after each iteration and compute the root-mean-square-error. Compare the error against the answer in part (a).

Answer:

The inverse of the covariance matrix is

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} 5.4545 & -3.6364 \\ -3.6364 & 9.0909 \end{bmatrix}$$

The results after each iteration are shown below:

Iteration	$\hat{\mu}_1$	$\hat{\mu}_2$	Imputed $x_{4,1}$	Imputed $x_{11,2}$	RMSE
1	0.5819	0.4962	0.2315	0.9301	0.1390
2	0.5527	0.5324	0.1826	0.9928	0.1683
3	0.5486	0.5376	0.1756	1.0018	0.1736
4	0.5480	0.5384	0.1746	1.0030	0.1743
5	0.5479	0.5385	0.1745	1.0032	0.1745

The root-mean-square-error for EM algorithm is considerably lower than that using mean imputation.

8. The purpose of this exercise is to illustrate the relationship between PCA and SVD. Let $\mathbf{A}$ be an $N \times d$ rectangular data matrix and $\mathbf{C}$ be its $d \times d$ covariance matrix.

   (a) Suppose $\mathbf{I}_N$ is an $N \times N$ identity matrix and $\mathbf{1}_N$ is an $N \times N$ matrix whose elements are equal to 1, i.e., $\forall i, j : (\mathbf{1})_{ij} = 1$. Show that the covariance matrix $\mathbf{C}$ can be expressed into the following form:

$$\mathbf{C} = \frac{1}{N-1}\mathbf{A}^T\left[\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\right]\mathbf{A}$$

   .

Answer: The covariance between columns i and j in matrix $\mathbf{A}$ is given by

$$C_{ij} = \frac{\sum_k (A_{ki} - \overline{A}_i)(A_{kj} - \overline{A}_j)}{N-1}, \tag{2.3}$$

where $\overline{A}_i$ and $\overline{A}_j$ are their corresponding column averages. A matrix of column averages for $\mathbf{A}$ can be computed as follows:

$$\begin{aligned}
\frac{1}{N}\mathbf{1}_N\mathbf{A} &= \frac{1}{N}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1d} \\ A_{21} & A_{22} & \cdots & A_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ A_{N1} & A_{N2} & \cdots & A_{Nd} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{N}\sum_j A_{j1} & \frac{1}{N}\sum_j A_{j2} & \cdots & \frac{1}{N}\sum_j A_{jd} \\ \frac{1}{N}\sum_j A_{j1} & \frac{1}{N}\sum_j A_{j2} & \cdots & \frac{1}{N}\sum_j A_{jd} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{N}\sum_j A_{j1} & \frac{1}{N}\sum_j A_{j2} & \cdots & \frac{1}{N}\sum_j A_{jd} \end{pmatrix}
\end{aligned} \tag{2.4}$$

Thus, each term $(A_{ki} - \overline{A}_i)$ in Equation (2.3) can be expressed in matrix notation as $A_{ki} - \frac{1}{N}\sum_j A_{ji} = [\mathbf{A} - \frac{1}{N}\mathbf{1}_N\mathbf{A}]_{ki}$. The covariance matrix $\mathbf{C}$ can therefore be computed as follows:

$$\begin{aligned}
\mathbf{C} &= \frac{1}{N-1}(\mathbf{A} - \frac{1}{N}\mathbf{1}_N\mathbf{A})^T(\mathbf{A} - \frac{1}{N}\mathbf{1}_N\mathbf{A}) \\
&= \frac{1}{N-1}\Big[(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)\mathbf{A}\Big]^T\Big[(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)\mathbf{A}\Big] \\
&= \frac{1}{N-1}\mathbf{A}^T\Big(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\Big)\Big(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\Big)\mathbf{A}
\end{aligned} \tag{2.5}$$

where we have use the following property of matrix transpose $(\mathbf{XY})^T = \mathbf{Y}^T\mathbf{X}^T$ on the last line. Furthermore, since the identity matrix and the matrix of all ones are symmetric, i.e., $\mathbf{I}_N^T = \mathbf{I}_N$ and $\mathbf{1}_N^T = \mathbf{1}_N$, therefore $(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)^T = (\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)$. Finally, it can be shown that the matrix $(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)$ is idempotent, which means it is the

same as the square of the matrix:

$$\begin{aligned}(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N)(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N) &= \mathbf{I}_N - \frac{2}{N}\mathbf{1}_N + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N \\ &= \mathbf{I}_N - \frac{2}{N}\mathbf{1}_N + \frac{1}{N^2}N\mathbf{1}_N \\ &= \mathbf{I}_N - \frac{2}{N}\mathbf{1}_N + \frac{1}{N}\mathbf{1}_N \\ &= \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N, \end{aligned} \tag{2.6}$$

where $\mathbf{1}_N\mathbf{1}_N = N\mathbf{1}_N$ is an $N \times N$ matrix whose elements are equal to N. Substituting (2.6) into (2.5), we obtain:

$$\mathbf{C} = \frac{1}{N-1}\mathbf{A}^T\left(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\right)\mathbf{A} \tag{2.7}$$

(b) Using singular value decomposition, the matrix $\mathbf{A}$ can be factorized as follows: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ is the $N \times N$ left singular matrix, $\mathbf{\Sigma}$ is the $N \times d$ matrix containing the singular values, and $\mathbf{V}$ is the $d \times d$ right singular matrix. Similarly, using eigenvalue decomposition, the covariance matrix can be factorized as follows: $\mathbf{C} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^T$. Show the relationship between SVD and PCA is given by the following equation:

$$\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T - \frac{1}{N}\mathbf{A}^T\mathbf{1}_N\mathbf{A} = (N-1)\mathbf{X}\mathbf{\Lambda}\mathbf{X}^T.$$

Answer: From the previous question, we can write:

$$\mathbf{C} = \frac{1}{N-1}\mathbf{A}^T\left(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\right)\mathbf{A} = \frac{1}{N-1}\left(\mathbf{A}^T\mathbf{A} - \frac{1}{N}\mathbf{A}^T\mathbf{1}_N\mathbf{A}\right) \tag{2.8}$$

Since $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{U}$ is an orthogonal matrix,

$$\mathbf{A}^T\mathbf{A} = [\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T]^T[\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T] = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T.$$

If $N > d$, then Σ has $N-d$ rows of all zeros. If we remove such rows, Σ becomes a $d \times d$ square matrix and $\Sigma^T\Sigma = \Sigma^2$. By substituting $\mathbf{C} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^T$ and $\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}$ into Equation (2.8), we have:

$$\mathbf{X}\mathbf{\Lambda}\mathbf{X}^T = \frac{1}{N-1}\left[\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T - \frac{1}{N}\mathbf{A}\mathbf{1}_N\mathbf{A}\right].$$

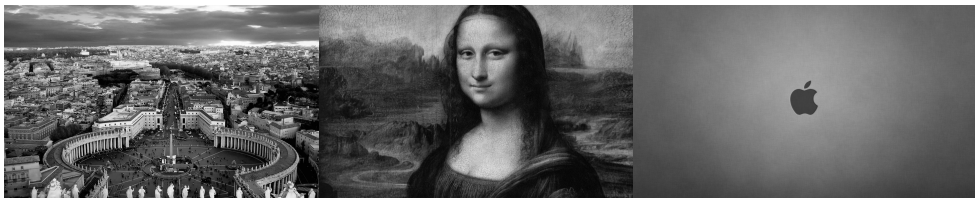
(c) Find the relationship between the right singular matrix $\mathbf{V}$ and the matrix of principal components $\mathbf{X}$ if the data matrix $\mathbf{A}$ has been column-centered (i.e., every column of $\mathbf{A}$ has been subtracted by the column mean) before applying SVD.

Answer: If the matrix $\mathbf{A}$ has been column-centered, then its column mean is zero, which means $\mathbf{A}^T\mathbf{1}_N$ is a matrix of all zeros. Thus, the last equation in the previous question reduces to:

$$\mathbf{X}\mathbf{\Lambda}\mathbf{X}^T = \frac{1}{N-1}\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T.$$

This suggests that the right singular matrix $\mathbf{V}$ corresponds to the principal components $\mathbf{X}$, while the square root of the singular values are the same as $N-1$ times the eigenvalues.

9. Principal component analysis (PCA) can be used for image compression by transforming a high-resolution image into its lower rank approximation. In this exercise, you will be provided with the following three images of size 1080 $\times$ 1920 pixels each (the filenames are `img1.jpg`, `img2.jpg`, and `img3.jpg`).



(a) img1 (b) img2 (c) img3

Figure 2.1. Image data set.

You will use Matlab to apply PCA to each of the following images.

(a) Load each image using the `imread` command. For example:

```
matlab> A = imread('img1.jpg');
```

(b) Plot the image in gray scale.

```
matlab> imagesc(A);
matlab> colormap(gray);
```

Answer: See Figure 2.1.

(c) Apply principal component analysis to obtain a reduced rank approximation of the image.

For example, to obtain a rank-10 approximation (i.e., using the first 10 principal components), use the following commands:

```
matlab> A = double(A);        % convert A from uint8 to double format
matlab> [U,V] = princomp(A);  % apply principal component analysis
matlab> rank = 10;            % set rank to be 10
matlab> B = V(:,1:rank)*U(:,1:rank)';   % B is the compressed image of A
matlab> figure;
matlab> imagesc(B);
matlab> colormap(gray);
```

For each image, vary the rank (i.e., number of principal components) as follows: 10, 30, 50, and 100. Save each image as follows:

```
matlab> saveas(gcf, 'filename.jpg', 'jpeg');
```

Insert the compressed (reduced rank) images to the solution file of your homework (don't submit the jpg files individually).

Answer: See Figure 2.2.

(d) Compare the size of matrix A (in bytes) to the total sizes of matrices U and V (in bytes). Compute the compression ratio:

$$\text{Compression ratio} = \frac{\text{Size of matrix A}}{\text{Size of matrix U} + \text{Size of matrix V}}$$

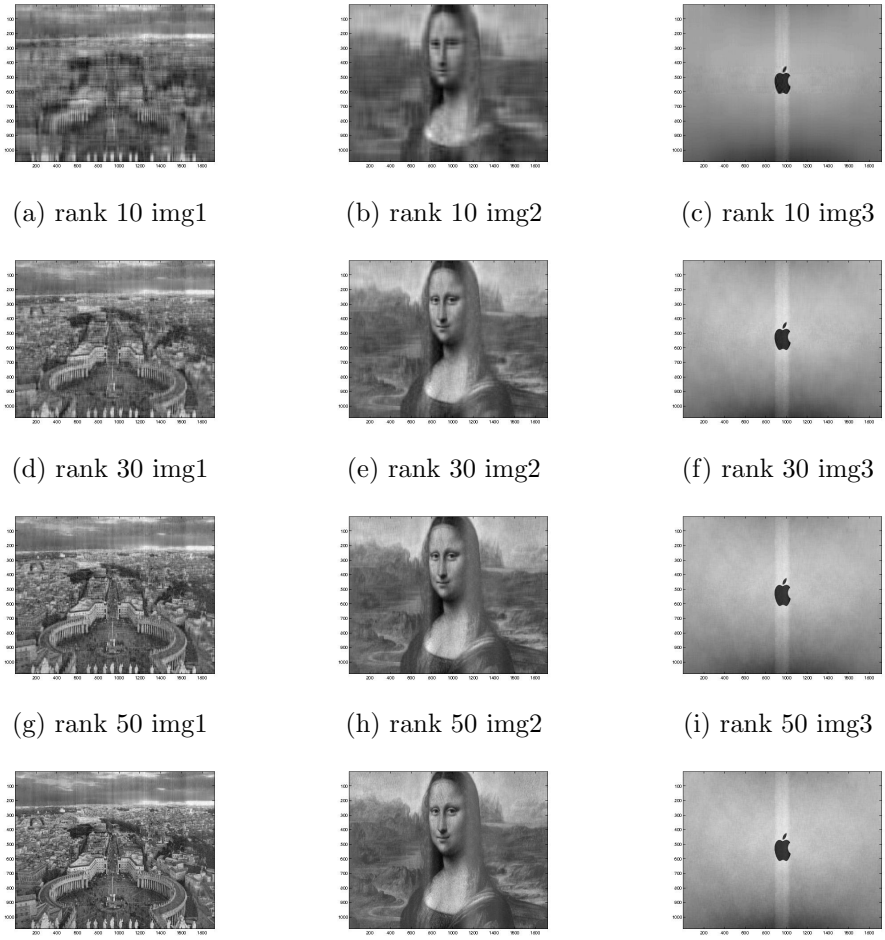
for each reduced rank (10, 30, 50, 100) of the images. You can use the `whos` command to determine the size of the matrices:

```
matlab> whos A U V
```

Answer: See Table 2.1.

rank	size of A	size of U	size of V	compression rate
10	16588800	153600	86400	69.12
30	16588800	460800	259200	23.04
50	16588800	768000	432000	13.824
100	16588800	1536000	864000	6.912

Table 2.1. Compression ratio for various images



(a) rank 10 img1 (b) rank 10 img2 (c) rank 10 img3

(d) rank 30 img1 (e) rank 30 img2 (f) rank 30 img3

(g) rank 50 img1 (h) rank 50 img2 (i) rank 50 img3

(j) rank 100 img1 (k) rank 100 img2 (l) rank 100 img3

Figure 2.2. Reduced-rank images using PCA

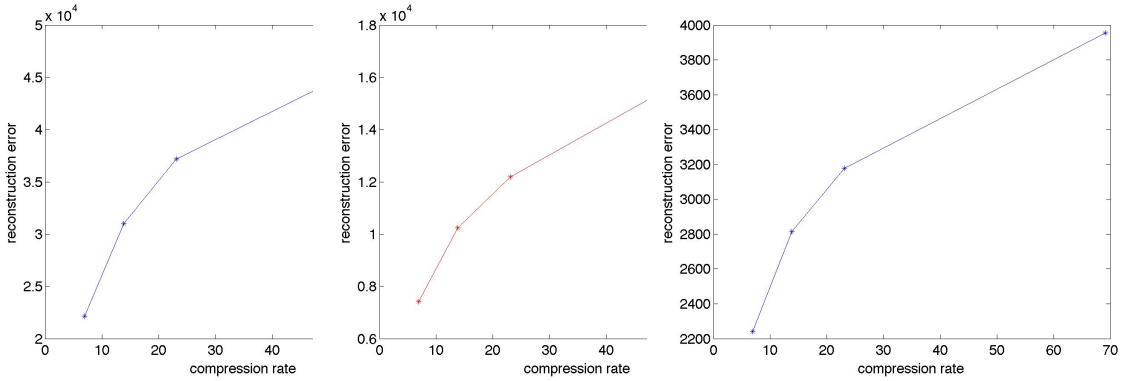
(e) Compute the reconstruction error $\|A - B\|_F$ of each reduced rank image, where $\|\cdot\|_F$ denote the Frobenius norm of a matrix. Note that the higher the reconstruction error, the lower the quality of the compressed image. Plot a graph of reconstruction error (y-axis) versus compression ratio (x-axis) for each image.

Answer: See Table 2.2 and Figure 2.3.

(f) State the minimum number of principal components (10, 30, 50, 100) needed to (visually) retain most of the salient features of each

image	rank	reconstruction error
img1	10	4.9565×10^4
img1	30	3.7198×10^4
img1	50	3.0998×10^4
img1	100	2.2135×10^4
img2	10	1.7798×10^4
img2	30	1.2190×10^4
img2	50	1.0236×10^4
img2	100	7.4063×10^3
img3	10	3.9544×10^3
img3	30	3.1775×10^3
img3	50	2.8146×10^3
img3	100	2.2397×10^3

Table 2.2. Reconstruction error for various images



(a) img1 (b) img2 (c) img3

Figure 2.3. Reconstruction error versus compression ratio

image (i.e., the city square in `img1.jpg`, shape of the face in `img2.jpg`, and shape of the apple in `img3.jpg`). Which image requires the least number of principal components? Which image requires the most number of principal components?

Answer:

img1.jpg: 50 components
img2.jpg: 30 components
img3.jpg: 10 components

2.3 Measures of Similarity and Dissimilarity

1. Consider the following binary vectors:

   $\mathbf{x_1} = (1, 1, 1, 1, 1)$

   $\mathbf{x_2} = (1, 1, 1, 0, 0)$

   $\mathbf{y_1} = (0, 0, 0, 0, 0)$

   $\mathbf{y_2} = (0, 0, 0, 1, 1)$

   (a) According to Jaccard coefficient, which pair of vectors—$(\mathbf{x}_1, \mathbf{x}_2)$ or $(\mathbf{y}_1, \mathbf{y}_2)$—are more similar to each other?

   Answer:

   $\text{Jaccard}(\mathbf{x}_1, \mathbf{x}_2) = \frac{3}{5} = 0.6$.

   $\text{Jaccard}(\mathbf{y}_1, \mathbf{y}_2) = \frac{0}{5} = 0$.

   Therefore, according to Jaccard coefficient, $(\mathbf{x}_1, \mathbf{x}_2)$ are more similar.

   (b) According to simple matching coefficient, which pair of vectors—$(\mathbf{x}_1, \mathbf{x}_2)$ or $(\mathbf{y}_1, \mathbf{y}_2)$—are more similar to each other?

   Answer:

   $\text{SMC}(\mathbf{x}_1, \mathbf{x}_2) = \frac{3}{5} = 0.6$.

   $\text{SMC}(\mathbf{y}_1, \mathbf{y}_2) = \frac{3}{5} = 0.6$.

   Therefore, according to simple matching coefficient, they are both equally similar.

   (c) According to Euclidean distance, which pair of vectors—$(\mathbf{x}_1, \mathbf{x}_2)$ or $(\mathbf{y}_1, \mathbf{y}_2)$—are more similar to each other?

   Answer:

   $\text{Euclidean}(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{2} = 1.4142$.

   $\text{Euclidean}(\mathbf{y}_1, \mathbf{y}_2) = \sqrt{2} = 1.4142$.

   Therefore, according to Euclidean distance, they are both equally similar.

2. Consider a weighted, undirected, graph G (see Figure 2.4 as an example). Let $e(u, v)$ be the weight of the edge between nodes u and v, where $e(u, u) = 0$ and $e(u, v) = \infty$ if u and v is disconnected. Assume the

graph is a connected component, i.e., there exists a path between every two nodes. Suppose the path length, $d(u, v)$, is defined as follows:

$$d(u,v) = \begin{cases} 0 & \text{if } u = v; \\ e(u,v), & \text{if there is an edge between } u \text{ and } v; \\ \min_{w \neq u \neq v} d(u,w) + d(w,v), & \text{otherwise.} \end{cases}$$

Is $d(u, v)$ a metric? State your reasons clearly. (Check whether the positivity, symmetry, and triangle inequality properties are preserved.).

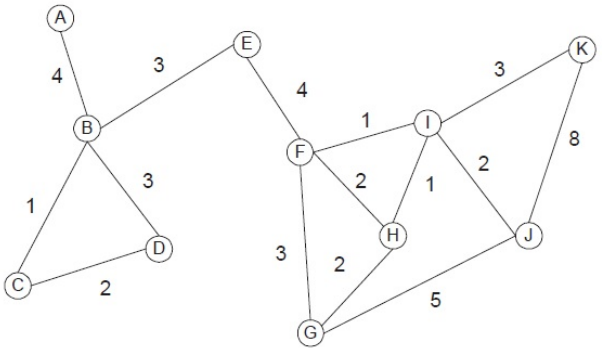


Figure 2.4. Weighted undirected graph.

Answer:

(a) Positivity property is preserved by definition since $d(u, u) = 0$ and $d(u, v) > 0$ if $u \neq v$.

(b) Symmetry property is preserved since the graph is undirected.

(c) Triangle inequality is not preserved. A counter-example is $d(K, J) \geq d(K, I) + d(I, J)$.

Therefore $d(u, v)$ is not a metric.

3. For document analysis, numerous measures have been proposed to determine the *semantic similarity* between two words using a domain ontology such as WordNet. For example, words such as `dog` and `cat` have higher semantic similarity than `dog` and `money` (since the former refers to two types of carnivores). Figure 2.5 below shows an example for computing the Wu-Palmer similarity between `dog` and `cat` based on their path

length in the WordNet hypernym hierarchy. The depth h refers to the length of the shortest path from the root to their lowest common hypernym (e.g., `carnivore` for the word pair `dog` and `cat`), whereas k is the minimum path length between the two words.

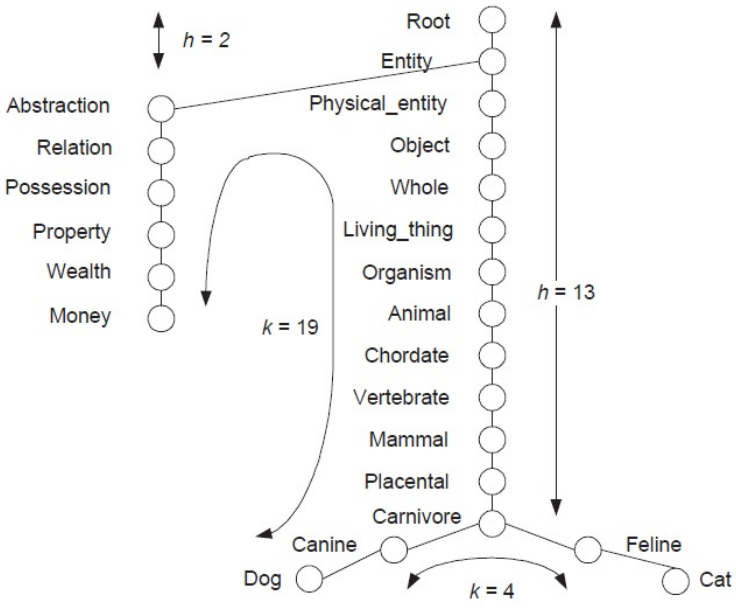


Figure 2.5. Sample of the hypernym hierarchy in WordNet.

The Wu-Palmer similarity measure is defined as follows:

$$W = \frac{2h}{k + 2h}$$

For example[1], for `dog` and `cat`, $W = 26/(4 + 26) = 0.867$, whereas for `dog` and `money`, $W = 4/(19 + 4) = 0.174$.

(a) What is the maximum and minimum possible value for Wu-Palmer similarity?

[1]In this simplified example, we assume each word has exactly 1 sense. In general, a word can have multiple senses. As a result, the Wu-Palmer measure is given by the highest similarity that can be achieved using one of its possible senses.

Answer: Maximum value is 1; minimum value approaches 0.

(b) Let $1 - W$ be the Wu-Palmer distance measure.

- Does $1 - W$ satisfy the positivity property?
  Answer: Yes. Since $1 - W = \frac{k}{2h} = 0$ when $k = 0$, this implies that $d(u, v) = 0$ if and only if $u = v$.
- Does $1 - W$ satisfy the symmetry property?
  Answer: Yes because W is a symmetric measure.
- Does $1 - W$ satisfy the triangle inequality property?
  Answer: No because each node can have more than one path to the root, some maybe shorter than others. For example, the words (money, statute) are very dissimilar to each other. But (money, bill) and (bill, statute) are very similar, thus violating triangle inequality. The actual path for these words in the WordNet ontology are shown in Figure 2.6.

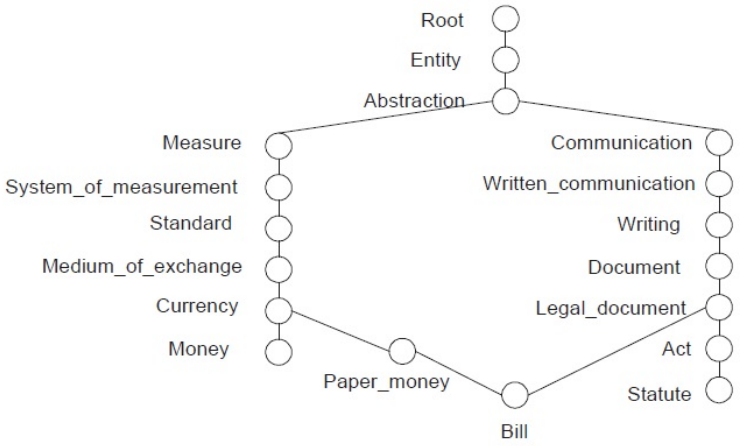


Figure 2.6. Sample of the hypernym hierarchy in WordNet.

4. Suppose you are given a census data, where every data object corresponds to a household and the following continuous attributes are used to characterize each household: total household income, number of household residents, property value, number of bedrooms, and number of vehicles owned. Suppose we are interested in clustering the households based on these attributes.

(a) Explain why cosine is not a good measure for clustering the data.

Answer: These attributes are all numerical and can have widely varying ranges of values, depending on the scale used to measure them. As a result, cosine measure will be biased by the attributes with largest range of magnitudes (e.g., total household income and property value).

(b) Explain why correlation is not a good measure for clustering the data.

Answer: The same argument as part (a). Because each attribute has different range, correlating the data points is meaningless.

(c) Explain what preprocessing steps and corresponding proximity measure you should use to do the clustering.

Answer: Euclidean distance, applied after standardizing the attributes to have a mean of 0 and a standard deviation of 1, would be appropriate

5. Consider the following distance measure:

$$d(\mathbf{x}, \mathbf{y}) = 1 - c(\mathbf{x}, \mathbf{y}),$$

where $c(\mathbf{x}, \mathbf{y})$ is the cosine similarity between two data objects, $\mathbf{x}$ and $\mathbf{y}$. Does the distance measure satisfy the positivity, symmetry, and triangle inequality properties? For each property, show your steps clearly. Assume $\mathbf{x}$ and $\mathbf{y}$ are non-negative vectors (e.g., term vectors for a pair of documents).

Answer:

(a) **Positivity** You need to show that $\forall \mathbf{x}, \mathbf{y} : d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \geq 0$ and $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$.

By definition, $\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$, where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$. Since $\cos\theta \leq 1$ (from trigonometry), therefore

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = 1 - \frac{\|\mathbf{x}\| \|\mathbf{y}\| \cos\theta}{\|\mathbf{x}\| \|\mathbf{y}\|} = 1 - \cos\theta \geq 0,$$

which completes the first part of the proof.

If $\mathbf{x} = \mathbf{y}$, then

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{x}}{\|\mathbf{x}\| \|\mathbf{x}\|} = 1 - \frac{\|\mathbf{x}\| \|\mathbf{x}\| \cos 0}{\|\mathbf{x}\| \|\mathbf{x}\|} = 0.$$

However, if $d(\mathbf{x}, \mathbf{y}) = 0$, then

$$1 - \cos\theta = 0 \Rightarrow \cos\theta = 1 \Rightarrow \theta = 0$$

In other words, as long as $\mathbf{x}$ and $\mathbf{y}$ are co-linear to each other, $d(\mathbf{x}, \mathbf{y}) = 0$ (even though $\mathbf{x} \neq \mathbf{y}$). The distance measure therefore does not satisfy the positivity property.

(b) **Symmetry**

Because $\mathbf{x} \cdot \mathbf{y} = \mathbf{y} \cdot \mathbf{x}$,

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{\mathbf{y} \cdot \mathbf{x}}{\|\mathbf{y}\|\|\mathbf{x}\|} = d(\mathbf{y}, \mathbf{x})$$

Hence, the distance measure satisfies the symmetry property.

(c) **Triangle Inequality**

First, note that $\cos\theta$ decreases with increasing θ for $0 \leq \theta \leq \pi/2$ (we focus only on this range of values for θ because the vectors are non-negative). Since the distance measure $d(\mathbf{x}, \mathbf{y}) = 1 - \cos\theta$ depends on the angle between the two vectors $\mathbf{x}$ and $\mathbf{y}$, the larger the angle, the larger the distance. We can show that the distance measure violates triangle inequality by choosing the angles in such a way that $d(\mathbf{x}, \mathbf{z}) > d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$. Consider the situation shown in Figure 2.7 below.

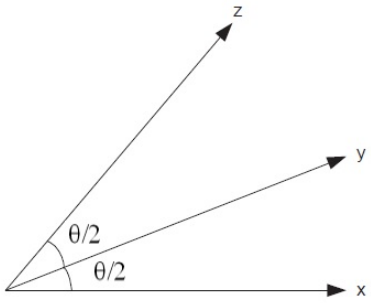


Figure 2.7. Triangle inequality violation example

In this case, we have: $d(\mathbf{x}, \mathbf{z}) = 1 - \cos\theta$ and $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{z}) = 1 - \cos(\theta/2)$. From trigonometry identities, $\cos\theta = 2\cos^2(\theta/2) -$

1. Therefore, $d(\mathbf{x}, \mathbf{z}) = 1 - \cos\theta = 1 - 2\cos^2(\theta/2) + 1 = 2 - 2\cos^2(\theta/2)$. On the other hand, $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) = 2 - 2\cos(\theta/2)$. Since $\cos^2(\theta/2) < \cos(\theta/2)$ as long as $0 < \cos(\theta/2) < 1$, we have found a counter-example where

$$d(\mathbf{x}, \mathbf{z}) = 2 - 2\cos^2(\theta/2) > d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) = 2 - 2\cos(\theta/2).$$

Here's a simple example. Suppose we have 3 documents and 2 words `data` and `mining`. Document $\mathbf{x}$ contains the word `data` only and document $\mathbf{z}$ contains the word `mining` only. However, document $\mathbf{y}$ contains both words. We can represent the documents as follows:

$$\mathbf{x} = (1, 0), \quad \mathbf{y} = (1, 1), \quad \mathbf{z} = (0, 1).$$

In this case, $d(\mathbf{x}, \mathbf{z}) = 1$ and $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{z}) = 1 - 1/\sqrt{2} = 0.2929$. Therefore, $d(\mathbf{x}, \mathbf{z}) > d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$, which is a violation of triangle inequality.

6. Consider a database of web graphs. Each graph is unweighted and contains a set of nodes and directed edges. A node corresponds to a web page while an edge is a transition from one page to another when a user clicks on a hyperlink or enters the URL directly into the location bar of the Web browser. Each web graph also represents the Web session of a user. Consider the following approaches for defining the similarity between two Web sessions, s_1 and s_2.

Approach 1 Node-based similarity

$$\text{Sim}_n(s_1, s_2) = \frac{\sum_i I(w_i \in s_1) \times I(w_i \in s_2)}{\max(|s_1|, |s_2|)},$$

where w_i is a web page, $|s_i|$ is the number of web pages visited during session s_i, $\max(a, b)$ is a function that returns the maximum value between a and b, and $I(w_i \in s_j)$ is an indicator function whose value is 1 if session s_j visited web page w_i and 0 otherwise.

Approach 2 Link-based similarity

$$\text{Sim}_l(s_1, s_2) = \frac{\sum_{i,j} I(w_i \to w_j \in s_1) \times I(w_i \to w_j \in s_2)}{\max(|s_1|, |s_2|)},$$

where $w_i \rightarrow w_j$ is a transition from page w_i to w_j, $|s_i|$ is the number of transitions in session s_i, $\max(a, b)$ is a function that returns the maximum value between a and b, and $I(w_i \rightarrow w_j \in s_k)$ is an indicator function whose value is 1 if session s_k contains a transition from web page w_i to w_j and 0 otherwise.

(a) Consider the following two Web sessions: $s_1 = (A \rightarrow B \rightarrow C \rightarrow B \rightarrow D \rightarrow E)$ and $s_2 = (A \rightarrow C \rightarrow B \rightarrow E)$. Compute the node-based and link-based similarities for the Web graphs constructed from the two sessions.

Answer: $\text{Sim}_n(s_1, s_2) = 4/5$ and $\text{Sim}_l(s_1, s_2) = 1/5$.

(b) Suppose the node-based similarity for s_1 and s_2 equals to 1. Can the web graphs for s_1 and s_2 be different? State your reasons clearly.

Answer: Yes. As long as both graphs contain the same set of nodes, the node-based similarity is equal to 1. But the graphs may still be different because the links in the graph could be different.

(c) Suppose $\text{Sim}_l(s_1, s_2) = 1$ according to approach 2. Can the web graphs for s_1 and s_2 be different? State your reasons clearly.

Answer: No. The web graphs are the same because all the node transitions in s_1 must also be present in s_2, and vice-versa.

(d) Which approach do you think is more effective at measuring similarity between two web sessions? State your reasons clearly.

Answer: Link-based similarity is more effective because its value is 1 only if the web graphs are isomorphic.

7. Consider the following distance measure $\mathcal{D}$ between two clusters of data points, $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbf{X}, \mathbf{y} \in \mathbf{Y}\},$$

where $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two data points, $\mathbf{x}$ and $\mathbf{y}$. Intuitively, $\mathcal{D}$ measures the distance between clusters in terms of the closest two points from each cluster (see Figure 2.8). Does the distance measure satisfy the positivity, symmetry, and triangle inequality properties? For each property, show your proof clearly or give a counter-example if the property is not satisfied.

Answer:

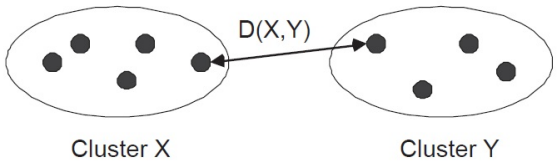


Figure 2.8. Cluster distance measure

(a) **Positivity:** Since Euclidean distance between any two data points is always non-negative, therefore $D(X, Y) \geq 0$. $D(X, y)$ can be zero even when $X \neq Y$ only if there is a data point is assigned to both clusters X and Y (i.e., if overlapping clusters are allowed). So, the distance measure satisfies the positivity property for disjoint clusters but not for overlapping clusters.

(b) **Symmetry:** Since Euclidean distance is a symmetric measure, $\mathcal{D}(\mathbf{X}, \mathbf{Y}) = \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbf{X}, \mathbf{y} \in \mathbf{Y}\} = \min\{d(\mathbf{y}, \mathbf{x}) : \mathbf{x} \in \mathbf{X}, \mathbf{y} \in \mathbf{Y}\} = \mathcal{D}(\mathbf{Y}, \mathbf{X})$. Thus, the measure is symmetric.

(c) **Triangle Inequality:** Triangle inequality property can be violated. A counter-example is shown in Figure 2.9.

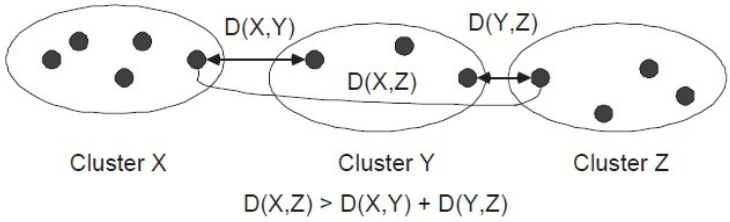


Figure 2.9. Violation of triangle inequality

8. For this question, assume each object is characterized by a set of continuous-valued attributes.

   (a) If two objects have a cosine similarity of 1, must their attribute values be identical? Explain.

Answer: No. A cosine similarity of 1 simply implies that the two attribute vectors are parallel to each other. For example, when $x = (1, 2)$ and $y = (2, 4)$, then their cosine similarity is 1.

(b) If two objects have a correlation value of 1, must their attribute values be identical? Explain.

Answer: No. A correlation value of 1 simply implies that there is a linear relationship between the two attribute vectors. For example, when $x = (1, 2)$ and $y = (3, 5)$, then their correlation is 1.

(c) If two objects have a Euclidean distance of 0, must their attribute values be identical? Explain.

Answer: Yes. Consider a pair of objects with attribute vectors x and y. Suppose their Euclidean distance is $d(x, y) = sqrt\sum_i (x_i - y_i)^2 = 0$, which is true only if $x_i = y_i$ for all i.

(d) Let $\mathbf{x}$ and $\mathbf{y}$ be the attribute vectors of two objects. State whether the following proximity measures—cosine, correlation, and Euclidean distance—are invariant (unchanged) under the following transformations. Specifically, if $x \rightarrow x'$ and $y \rightarrow y'$, would cosine(x, y) = cosine(x', y'), correlation(x, y) = correlation(x', y'), and Euclidean(x, y) = Euclidean(x', y')?

i. Translation: $\mathbf{x} \rightarrow \mathbf{x} + c$ and $\mathbf{y} \rightarrow \mathbf{y} + c$, where c is a constant added to each attribute value in $\mathbf{x}$ and $\mathbf{y}$.
Answer: Cosine is not invariant because cosine$(\mathbf{x}+c, \mathbf{y}+c) = \frac{\sum_i (x_i+c)(y_i+c)}{\sqrt{\sum_i (x_i+c)^2 (y_i+c)^2}} \neq \frac{\sum_i x_i y_i}{\sqrt{x_i^2 y_i^2}}$ unless $c = 0$. Euclidean distance is invariant since $\sqrt{\sum_i [(x_i + c) - (y_i + c)]^2} = \sqrt{\sum_i (x_i - y_i)^2}$. Similarly, correlation measure is also invariant because when $\mathbf{x} \rightarrow \mathbf{x} + c$, then the mean will also be shifted $\overline{x} \rightarrow \overline{x} + c$ but the standard deviation remains unchanged since $\sigma_x = \sqrt{\sum_i (x_i + c - \overline{x} - c)^2} = \sqrt{\sum_i (x_i - \overline{x})^2}$. Thus, correlation$(\mathbf{x} + c, \mathbf{y}+c) = \frac{\sum_i (x_i+c-\overline{x}-c)(y_i+c-\overline{y}-c)}{\sigma_x \sigma_y} = \frac{\sum_i (x_i-\overline{x})(y_i-\overline{y})}{\sigma_x \sigma_y}$ = correlation(**x**,**y**).

ii. Scaling: $\mathbf{x} \rightarrow c\mathbf{x}$ and $\mathbf{y} \rightarrow c\mathbf{y}$, where c is a constant multiplied to each attribute value in $\mathbf{x}$ and $\mathbf{y}$.
Answer: Cosine is invariant because $\frac{\sum_i c x_i c y_i}{\sqrt{(\sum_i c x_i)^2 (\sum_j c y_j)^2}} = \frac{\sum_i x_i y_i}{\sqrt{(\sum_i x_i)^2 (\sum_j y_j)^2}}$. Correlation is also invariant because when $\mathbf{x} \rightarrow c\mathbf{x}$, then both the mean and standard deviation are re-scaled by the same factor: $\overline{x}' = \frac{\sum_i c x_i}{n} = c\overline{x}$ and $\sigma_{x'} = \sqrt{\sum_i (c x_i - c\overline{x})^2} = c\sigma_x$. Euclidean distance is not invariant because $\sqrt{\sum_i (c x_i - c y_i)^2} = c\sqrt{\sum_i (x_i - y_i)^2}$.

iii. Standardization: $\mathbf{x} \rightarrow (\mathbf{x} - c)/d$ and $\mathbf{y} \rightarrow (\mathbf{y} - c)/d$, where c and d are constants.
Answer: Standardization is a combination of translation (by the mean of the vector) and scaling (by the standard deviation). Since correlation is invariant with respect to both operations, it is also invariant with respect to standardization. However, cosine and Euclidean distance are not invariant since they are not preserved by one of the two operations.

9. Consider the following survey data about users who joined an online community. The sample covariance between the user's height (in mm) and number of years being a member of the community is 5.0.

   (a) Suppose the sample covariance between the user's age and number of years being a member of the community is only 0.5. Does this imply that user's height is more correlated with number of years in the community than user's age? Answer yes or no and explain your reasons clearly.
   Answer: No. Covariance is not a dimensionless quantity, so its magnitude depends on the scale of measurement.

   (b) Suppose the height attribute is re-defined as height above the average for all users who participated in the survey. For example, a user who is 1650 mm tall has a height value of -50 mm (assuming the average height of all users is 1700 mm). Would the covariance between the re-defined height attribute and number of years in the community be greater than, smaller than, or equal to 5.0?
   Answer: Equal. Let x_h denote the height attribute and x_y be the number of years in the community. The sample covariance between the two attributes is given by:

$$\Sigma_{x_h,x_y} = \frac{1}{N-1}\sum_{N}^{i=1}(x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y}),$$

   where $\overline{x_h}$ and $\overline{x_y}$ are the average height and average number of years, respectively. If we re-define the height attribute as $x'_h = x_h - \overline{x_h}$,

then $\overline{x'_h} = 0$. Hence, the covariance between x'_h and x_y becomes

$$\begin{aligned}
\Sigma_{x'_h,x_y} &= \frac{1}{N-1}\sum_{i=1}^{N}(x'_{ih} - \overline{x'_h})(x_{iy} - \overline{x_y}) \\
&= \frac{1}{N-1}\sum_{i=1}^{N}(x_{ih} - \overline{x_h} - 0)(x_{iy} - \overline{x_y}) \\
&= \Sigma_{x_h,x_y}
\end{aligned}$$

This result means centering the height attribute has no effect on its covariance to other attributes.

(c) If the measurement unit for height is converted from mm to inches (where 1 inch = 25.4 mm), will the covariance between height (in inches) and number of years in the community be greater than, smaller than, or equal to 5.0?

Answer: Re-scaling the height attribute is equivalent to multiplying the original attribute by some constant C, i.e., $x'_h = Cx_h$. Furthermore, we can show that $\overline{x'_h} = C\overline{x_h}$. Thus the covariance between the rescaled height and number of years in the community will be:

$$\begin{aligned}
\Sigma_{x'_h,x_y} &= \frac{1}{N-1}\sum_{i=1}^{N}(x'_{ih} - \overline{x'_h})(x_{iy} - \overline{x_y}) \\
&= \frac{1}{N-1}\sum_{i=1}^{N}(Cx_{ih} - C\overline{x_h})(x_{iy} - \overline{x_y}) \\
&= \frac{C}{N-1}\sum_{i=1}^{N}(x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y}) \\
&= C\Sigma_{x_h,x_y}
\end{aligned}$$

In this case, $C = \frac{1}{25.4}$ which is smaller than 1. Therefore, the covariance value will be smaller when you convert the unit from mm to inches.

(d) Suppose you standardize both the height and number of years in the community attributes (by subtracting their respective means and dividing by their corresponding standard deviations). Would their covariance value be greater than, smaller than, or equal to

5.0? To obtain full credit, you must prove your answer by showing the computations clearly.

Answer: The re-defined attributes after standardization are: $x'_h = \frac{x_h - \overline{x_h}}{\sigma_h}$, $x'_y = \frac{x_y - \overline{x_y}}{\sigma_y}$. Furthermore, we can show that $\overline{x'_h} = 0$, $\overline{x'_y} = 0$. Then,

$$\begin{aligned}
\Sigma_{x'_h, x'_y} &= \frac{1}{N-1} \sum_{N}^{i=1} (x'_{ih} - \overline{x'_h})(x'_{iy} - \overline{x'_y}) \\
&= \frac{1}{N-1} \sum_{i=1}^{N} (\frac{x_{ih} - \overline{x_h}}{\sigma_h})(\frac{x_{iy} - \overline{x_y}}{\sigma_y}) \\
&= \frac{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y})}{\sigma_h \sigma_y} \\
&= \frac{\Sigma_{x_h, x_y}}{\sigma_h \sigma_y} \qquad (2.9)
\end{aligned}$$

Note that $\frac{1}{\sigma_h \sigma_y} \Sigma_{x_h, x_y}$ is equivalent to the correlation coefficient between x_h and x_y. Since correlation coefficient is always less than or equal to 1 whereas the original covariance value is +5, this means that the covariance value is smaller after standardization.

Next, we will prove that correlation coefficient is always between -1 and $+1$. First, note that

$$\sigma_h = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ih} - \overline{x}_h)^2}, \quad \sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{iy} - \overline{x}_y)^2}.$$

Thus, Equation (2.12) can be re-written as follows:

$$\begin{aligned}
\Sigma_{x'_h, x'_y} &= \frac{\Sigma_{x_h, x_y}}{\sigma_h \sigma_y} \\
&= \frac{\sum_{i=1}^{N} (x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y})}{\sqrt{\left(\sum_{i=1}^{N} (x_{ih} - \overline{x}_h)^2 \right) \left(\sum_{i=1}^{N} (x_{iy} - \overline{x}_y)^2 \right)}} \qquad (2.10)
\end{aligned}$$

Let $h_i = x_{ih} - \overline{x}_h$ and $y_i = x_{iy} - \overline{x}_y$. Equation (2.13) becomes

$$\Sigma_{x'_h,x'_y} = \frac{\sum_{i=1}^{N} h_i y_i}{\sqrt{\left(\sum_{i=1}^{N} h_i^2\right)\left(\sum_{i=1}^{N} y_i^2\right)}} = \frac{\vec{h} \bullet \vec{y}}{|\vec{h}||\vec{y}|} \quad (2.11)$$

According to Cauchy-Schwarz inequality, for any vectors $\vec{h}$ and $\vec{y}$, we have

$$\vec{h} \bullet \vec{y} \leq |\vec{h}||\vec{y}|.$$

Thus the ratio on the right-hand side of Equation (2.14) is less than or equal to 1, which completes the proof.

10. Suppose you are given a database of patient's demographic information from a healthcare provider. The covariance matrix obtained for three attributes: age, weight, and systolic blood pressure (bp) is shown below:

$$\begin{array}{r} age \rightarrow \\ weight \rightarrow \\ bp \rightarrow \end{array} \begin{pmatrix} 389.75 & 199.37 & 135.12 \\ 199.37 & 610.52 & 426.30 \\ 135.12 & 426.30 & 359.36 \end{pmatrix}$$

(a) Does this imply that user's age is more correlated with his/her weight than systolic blood pressure? Answer yes or no and explain your reasons clearly.

Answer: No. Covariance is not a dimensionless quantity, so its magnitude depends on the scale of measurement. Even though covariance between age and weight is higher than that between age and systolic blood pressure, it is possible the correlation is lower.

(b) Suppose the weight attribute is centered by subtracting it with the average weight of all patients in the database. For example, a 200-pound patient has a weight recorded as 50 (if the average weight of the patients is 150 pounds). Would the covariance between the centered weight attribute and age be greater than, smaller than, or equal to 199.37?

Answer: Equal. Let x_h denote the weight attribute and x_y is the age attribute. The sample covariance between the two attributes is given by:

$$\Sigma_{x_h,x_y} = \frac{1}{N-1} \sum_{N}^{i=1} (x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y}),$$

where $\overline{x_h}$ and $\overline{x_y}$ are the average weight and average age, respectively. If we re-define the weight attribute as $x'_h = x_h - \overline{x_h}$, then $\overline{x'_h} = 0$. Hence, the covariance between x'_h and x_y becomes

$$\begin{aligned}\Sigma_{x'_h,x_y} &= \frac{1}{N-1}\sum_{i=1}^{N}(x'_{ih} - \overline{x'_h})(x_{iy} - \overline{x_y}) \\ &= \frac{1}{N-1}\sum_{i=1}^{N}(x_{ih} - \overline{x_h} - 0)(x_{iy} - \overline{x_y}) \\ &= \Sigma_{x_h,x_y}\end{aligned}$$

This result means centering the weight attribute has no effect on its covariance to other attributes.

(c) If the measurement unit for weight is converted from pounds to kilograms (where 1 kg = 2.2 pounds), will the covariance between weight (in kilogram) and age be greater than, smaller than, or equal to 199.37?

Answer: Re-scaling the weight attribute is equivalent to multiplying the original attribute by some constant C, i.e., $x'_h = Cx_h$. Furthermore, we can show that $\overline{x'_h} = C\overline{x_h}$. Thus the covariance between the rescaled weight and age will be:

$$\begin{aligned}\Sigma_{x'_h,x_y} &= \frac{1}{N-1}\sum_{i=1}^{N}(x'_{ih} - \overline{x'_h})(x_{iy} - \overline{x_y}) \\ &= \frac{1}{N-1}\sum_{i=1}^{N}(Cx_{ih} - C\overline{x_h})(x_{iy} - \overline{x_y}) \\ &= \frac{C}{N-1}\sum_{i=1}^{N}(x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y}) \\ &= C\Sigma_{x_h,x_y}\end{aligned}$$

In this case, $C = \frac{1}{2.2}$ which is smaller than 1. Therefore, the covariance value will be smaller when you convert the unit from pounds to kilograms.

(d) Suppose you standardize both the age and weight attributes (by subtracting their respective means and dividing by their corresponding standard deviations). Would their covariance value be greater than, smaller than, or equal to 199.37?

Answer: The re-defined attributes after standardization are: $x'_h = \frac{x_h - \overline{x_h}}{\sigma_h}$, $x'_y = \frac{x_y - \overline{x_y}}{\sigma_y}$. Furthermore, we can show that $\overline{x'_h} = 0$, $\overline{x'_y} = 0$. Then,

$$\begin{aligned}
\Sigma_{x'_h, x'_y} &= \frac{1}{N-1} \sum_{N}^{i=1} (x'_{ih} - \overline{x'_h})(x'_{iy} - \overline{x'_y}) \\
&= \frac{1}{N-1} \sum_{i=1}^{N} (\frac{x_{ih} - \overline{x_h}}{\sigma_h})(\frac{x_{iy} - \overline{x_y}}{\sigma_y}) \\
&= \frac{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y})}{\sigma_h \sigma_y} \\
&= \frac{\Sigma_{x_h, x_y}}{\sigma_h \sigma_y} \qquad (2.12)
\end{aligned}$$

Note that $\frac{1}{\sigma_h \sigma_y} \Sigma_{x_h, x_y}$ is equivalent to the correlation coefficient between x_h and x_y. Since correlation coefficient is always less than or equal to 1 whereas the original covariance value is +5, this means that the covariance value is smaller after standardization.

Next, we will prove that correlation coefficient is always between -1 and $+1$. First, note that

$$\sigma_h = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ih} - \overline{x}_h)^2}, \quad \sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{iy} - \overline{x}_y)^2}.$$

Thus, Equation (2.12) can be re-written as follows:

$$\begin{aligned}
\Sigma_{x'_h, x'_y} &= \frac{\Sigma_{x_h, x_y}}{\sigma_h \sigma_y} \\
&= \frac{\sum_{i=1}^{N} (x_{ih} - \overline{x_h})(x_{iy} - \overline{x_y})}{\sqrt{\left(\sum_{i=1}^{N} (x_{ih} - \overline{x}_h)^2 \right) \left(\sum_{i=1}^{N} (x_{iy} - \overline{x}_y)^2 \right)}} \qquad (2.13)
\end{aligned}$$

Let $h_i = x_{ih} - \overline{x}_h$ and $y_i = x_{iy} - \overline{x}_y$. Equation (2.13) becomes

$$\Sigma_{x'_h, x'_y} = \frac{\sum_{i=1}^{N} h_i y_i}{\sqrt{\left(\sum_{i=1}^{N} h_i^2 \right) \left(\sum_{i=1}^{N} y_i^2 \right)}} = \frac{\mathbf{h}^T \mathbf{y}}{\|\mathbf{h}\| \|\mathbf{y}\|} \qquad (2.14)$$

According to Cauchy-Schwarz inequality, for any vectors $\mathbf{h}$ and $\mathbf{y}$, we have

$$\mathbf{h}^T\mathbf{y} \leq \|\mathbf{h}\|\|\mathbf{y}\|.$$

Thus the ratio on the right-hand side of Equation (2.14) is less than or equal to 1, which completes the proof.

11. Consider the following distance measure for two sets, $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = 1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|},$$

where $\cap$ is the intersection between the two sets, $\cap$ is the union of the two sets, and $|\cdot|$ denote the cardinality of the set. This measure is equivalent to 1 minus the Jaccard similarity. Does the distance measure satisfy the positivity, symmetry, and triangle inequality properties? For each property, explain your reason clearly or give a counter-example if the property is not satisfied.

Answer:

(a) Positivity: Since $|\mathbf{X} \cap \mathbf{Y}| \leq |\mathbf{X} \cup \mathbf{Y}|$, therefore $\frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} \leq 1$ and

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = 1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} \geq 0.$$

Furthermore, if $\mathbf{X} = \mathbf{Y}$, then $|\mathbf{X} \cup \mathbf{Y}| = |\mathbf{X} \cap \mathbf{Y}| = |\mathbf{X}|$. Hence,

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = 1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} = 1 - 1 = 0.$$

Similarly, if $\mathcal{D}(\mathbf{X}, \mathbf{Y}) = 0$, then

$$1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} = 0,$$

which means, $|\mathbf{X} \cup \mathbf{Y}| = |\mathbf{X} \cap \mathbf{Y}|$, or equivalently, $\mathbf{X} = \mathbf{Y}$.
Hence, the positivity property holds for the distance measure.

(b) Symmetry:

$$\mathcal{D}(\mathbf{X}, \mathbf{Y}) = 1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} = 1 - \frac{|\mathbf{Y} \cap \mathbf{X}|}{|\mathbf{Y} \cup \mathbf{X}|} = \mathcal{D}(\mathbf{Y}, \mathbf{X}).$$

Hence, the symmetry property holds for the distance measure.

(c) Triangle inequality:

$$\begin{aligned}\mathcal{D}(\mathbf{X},\mathbf{Y})+\mathcal{D}(\mathbf{Y},\mathbf{Z}) &= 1-\frac{|\mathbf{X}\cap\mathbf{Y}|}{|\mathbf{X}\cup\mathbf{Y}|}+1-\frac{|\mathbf{Y}\cap\mathbf{Z}|}{|\mathbf{Y}\cup\mathbf{Z}|}\\ &= \frac{|\mathbf{X}\cup\mathbf{Y}|-|\mathbf{X}\cap\mathbf{Y}|}{|\mathbf{X}\cup\mathbf{Y}|}+\frac{|\mathbf{Y}\cup\mathbf{Z}|-|\mathbf{Y}\cap\mathbf{Z}|}{|\mathbf{Y}\cup\mathbf{Z}|}\\ &\geq \frac{|\mathbf{X}\cup\mathbf{Y}|-|\mathbf{X}\cap\mathbf{Y}|}{|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|}+\frac{|\mathbf{Y}\cup\mathbf{Z}|-|\mathbf{Y}\cap\mathbf{Z}|}{|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|}\end{aligned}$$

and

$$\mathcal{D}(\mathbf{X},\mathbf{Z})=1-\frac{|\mathbf{X}\cap\mathbf{Z}|}{|\mathbf{X}\cup\mathbf{Z}|}\leq 1-\frac{|\mathbf{X}\cap\mathbf{Z}|}{|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|}=\frac{|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|-|\mathbf{X}\cap\mathbf{Z}|}{|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|}.$$

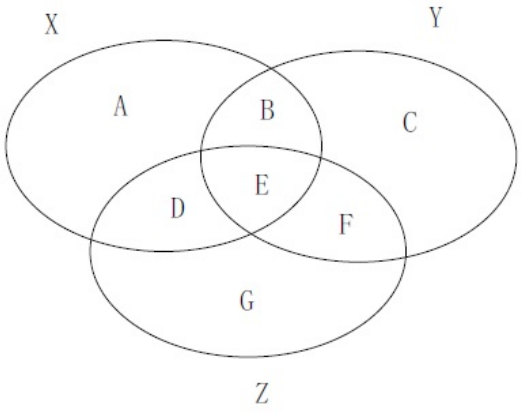


Figure 2.10. Illustration of triangle inequality

Figure 2.10 shows the Venn diagram for sets **X**, **Y** and **Z**. The number of data points in each subregion in the Venn Diagram is labeled **A** through **G**. From this figure, it can be easily seen that,
$|\mathbf{X}\cup\mathbf{Y}|-|\mathbf{X}\cap\mathbf{Y}|+|\mathbf{Y}\cup\mathbf{Z}|-|\mathbf{Y}\cap\mathbf{Z}| = \mathbf{A}+\mathbf{C}+\mathbf{D}+\mathbf{F}+\mathbf{B}+\mathbf{C}+\mathbf{D}+\mathbf{G}$
whereas
$|\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}|-|\mathbf{X}\cap\mathbf{Z}| = \mathbf{A}+\mathbf{B}+\mathbf{C}+\mathbf{F}+\mathbf{G}$.

The preceding equations suggest that

$$|\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}| - |\mathbf{X} \cap \mathbf{Z}| \leq |\mathbf{X} \cup \mathbf{Y}| - |\mathbf{X} \cap \mathbf{Y}| + |\mathbf{Y} \cup \mathbf{Z}| - |\mathbf{Y} \cap \mathbf{Z}|$$

Putting the inequalities together, we have

$$\begin{aligned} \mathcal{D}(\mathbf{X}, \mathbf{Z}) &\leq \frac{|\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}| - |\mathbf{X} \cap \mathbf{Z}|}{|\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}|} \\ &\leq \frac{|\mathbf{X} \cup \mathbf{Y}| - |\mathbf{X} \cap \mathbf{Y}| + |\mathbf{Y} \cup \mathbf{Z}| - |\mathbf{Y} \cap \mathbf{Z}|}{|\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}|} \\ &\leq \mathcal{D}(\mathbf{X}, \mathbf{Y}) + \mathcal{D}(\mathbf{Y}, \mathbf{Z}) \end{aligned}$$

12. Which similarity or distance measure is most effective for each of the domains given below:

    (a) Which measure, Jaccard or Simple Matching Coefficient, is most appropriate to compare how similar are the answers provided by students in an exam. Assume that the answers to all the questions in the exam are either True or False.

    Answer: Simple matching coefficient. The values of true and false are equally important when computing similarity.

    (b) Which measure, Jaccard or Simple Matching Coefficient, is most appropriate to compare how similar are the locations visited by tourists at an amusement park. Assume the location information is stored as binary yes/no attributes (yes means a location was visited by the tourist and no means a location has not been visited).

    Answer: Jaccard. Here places visited by the tourists should play a more significant role in computing similarity than places they did not visit.

    (c) Which measure, Euclidean distance or correlation coefficient, is most appropriate to compare two flows in a network traffic. For each flow, we record information about the number of packets transmitted, number of bytes transferred, number of acknowledgments sent, and duration of the session.

    Answer: Euclidean distance (after standardizing each attribute). Correlation coefficient is not meaningful here because it is not meaningful to correlate two flows which has different attribute values (i.e., correlating the attributes are meaningful but correlating the flows are not).

(d) Which measure, Euclidean distance or cosine similarity, is most appropriate to compare the coordinates of a moving object in a 2-dimensional space. For example, using GPS data, the object may be located at (31.4° West,12.4° North) at time t_1 and (29.4° West,12.5° North) at another time t_2. Note: we may use +/- to indicate East/West or North/South directions when computing the similarity or distance measures.

Answer: Euclidean distance. This is because cosine measures the angle of the two locations. Thus, if two locations lie along the same line through the origin, their cosine similarity will be 0 even though they are located far away from each other.

(e) Which measure, Euclidean distance or cosine similarity, is most appropriate to compare the similarity of items bought by customers at a grocery store. Assume each customer is represented by a 0/1 binary vector of items (where a 1 means the customer had previously bought the item).

Answer: Cosine similarity because presence of an item in the transaction plays a more important role in determining similarity than absence of the item.

3

Classification

3.1 Decision Tree

1. Consider a training set sampled uniformly from the the two-dimensional space shown in Figure 3.1.

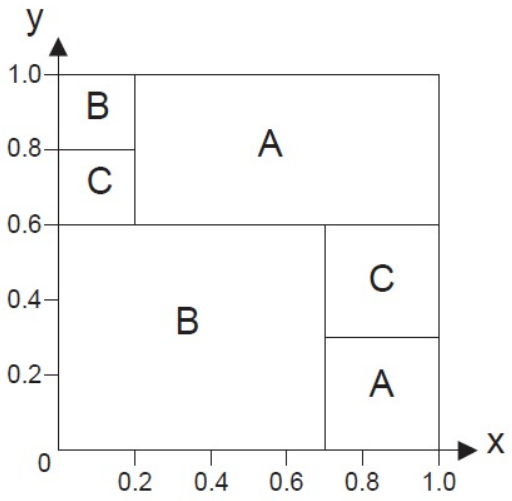


Figure 3.1. 2D region

Assume that the training set size is large enough so that the probabilities can be calculated accurately based on the areas of the selected regions.

The space is divided into three classes—A, B, and C. In this exercise, you will build a decision tree from the training set.

(a) Compute the entropy for the overall data.

Answer: For overall data, $p(A) = 0.32 + 0.09 = 0.41$, $p(B) = 0.42 + 0.04 = 0.46$, and $p(C) = 0.04 + 0.09 = 0.13$. Therefore the overall entropy is

$$-0.41 \log_2 0.41 - 0.46 \log_2 0.46 - 0.13 \log_2 0.13 = 1.4254.$$

(b) Compare the entropy when the data is split at $x <= 0.2$, $x <= 0.7$, and $y <= 0.6$.

Answer:

i. Split at $x = 0.2$:
For the child node $x \leq 0.2$, $p(A) = 0$, $p(B) = 0.8$, and $p(C) = 0.2$. Its entropy is $-0.8 \log_2 0.8 - 0.2 \log_2 0.2 = 0.7219$. For the child node $x > 0.2$, $p(A) = 0.41/0.80 = 0.5125$, $p(B) = 0.3/0.8 = 0.3750$, and $p(C) = 0.09/0.80 = 0.1125$. Its entropy is $-0.5125 \log_2 0.5125 - 0.375 \log_2 0.375 - 0.1125 \log_2 0.1125 = 1.3795$. Therefore, the average entropy for the children is $0.2 \times 0.7219 + 0.8 \times 1.3795 = 1.2480$.

ii. Split at $x = 0.7$:
For the child node $x \leq 0.7$, $p(A) = 0.2/0.7 = 0.2857$, $p(B) = 0.46/0.7 = 0.6571$, and $p(C) = 0.04/0.7 = 0.0571$. Its entropy is $-0.2857 \log_2 0.2857 - 0.6571 \log_2 0.6571 - 0.0571 \log_2 0.0571 = 1.1503$. For the child node $x > 0.7$, $p(A) = 0.7$, $p(B) = 0$, and $p(C) = 0.3$. Its entropy is $-0.7 \log_2 0.7 - 0.3 \log_2 0.3 = 0.8813$. Therefore the average entropy for the children is $0.7 \times 1.1503 + 0.3 \times 0.8813 = 1.0696$.

iii. Split at $y = 0.6$.
For the child node $y \leq 0.6$, $p(A) = 0.09/0.6 = 0.15$, $p(B) = 0.42/0.6 = 0.7$, and $p(C) = 0.09/0.6 = 0.15$. Its entropy is $-0.15 \log_2 0.15 - 0.7 \log_2 0.7 - 0.15 \log_2 0.15 = 1.1813$. For the child node $y > 0.6$, $p(A) = 0.32/0.4 = 0.8$ and $p(B) = p(C) = 0.04/0.4 = 0.1$. Its entropy is $-0.8 \log_2 0.8 - 0.1 \log_2 0.1 - 0.1 \log_2 0.1 = 0.9219$. Therefore, the average entropy for the children is $0.6 \times 1.1813 + 0.4 \times 0.9219 = 1.0776$.

(c) Based on your answer in part (b), which attribute split condition do you think should be used as the root of the decision tree.

Answer: Comparing their entropy values, the split at $x = 0.7$ has the highest gain.

(d) Draw the full decision tree for the data set.

Answer: Full decision tree:

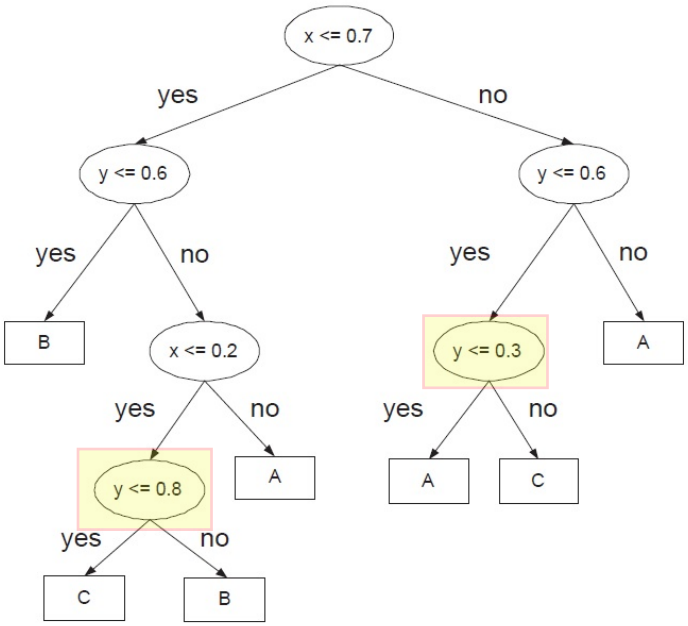


Figure 3.2. Decision tree for 2D region.

2. Draw the full decision tree that perfectly classifies each of the data sets given below. There could be more than one answer to each question (you only need to draw one). You do not have to consider the impurity measure used by the decision tree algorithm. Ignore pre-pruning and post-pruning. Assume there are no noise and missing attribute values.

   (a) Consider a data set with three Boolean attributes, A, B, and C, and a binary class label y whose value is True if the number of attributes with True values is even and False otherwise. For example, if A=True, B=True, C=False, then y=True (because there are two attributes with True values).

Answer: This corresponds to a parity function for 3 Boolean attributes.

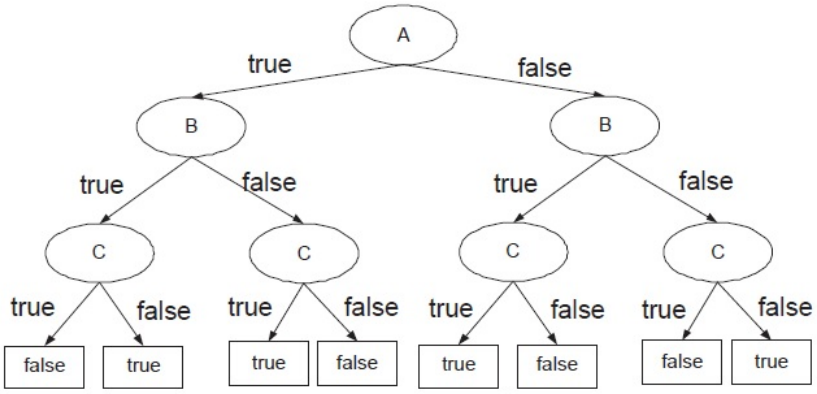


Figure 3.3. Decision tree for parity function

(b) Consider the 2-dimensional data set shown in Figure 3.4, where A, B, and C are the class labels for the respective regions.

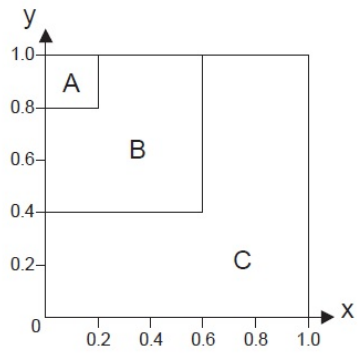


Figure 3.4. 2D region

Answer: The decision tree is shown in Figure 3.5.

3. Show that the gini index of a node never increases after it has been split into smaller successor nodes. To simplify the problem, you can assume

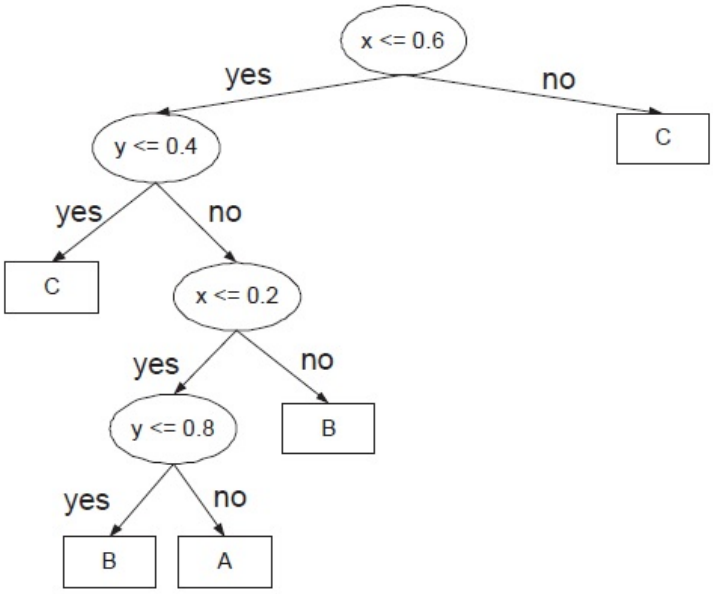


Figure 3.5. Decision tree for 2D region

that both the splitting attribute associated with the node and the class label are binary valued.

Answer:

Suppose there are N training examples, divided into 2 classes, with n_+ positive and n_- negative examples. The gini index before splitting is:

$$g_1 = 1 - \left(\frac{n_+}{N}\right)^2 - \left(\frac{n_-}{N}\right)^2 \tag{3.1}$$

After splitting on a binary attribute X, let n_{1+} and n_{1-} be the number of positive and negative examples associated with the left child whereas n_{2+} and n_{2-} be the number of positive and negative examples associated with the right child. Furthermore,

$$\begin{aligned} n_1 &= n_{1+} + n_{1-} \\ n_2 &= n_{2+} + n_{2-} \\ n_+ &= n_{1+} + n_{2+} \\ n_- &= n_{1-} + n_{2-} \\ n_1 + n_2 &= N \end{aligned}$$

The gini index after splitting is given as follows:

$$\begin{aligned} g_2 &= \frac{n_1}{N}\left[1-\left(\frac{n_{1+}}{n_1}\right)^2-\left(\frac{n_{1-}}{n_1}\right)^2\right]+\frac{n_2}{N}\left[1-\left(\frac{n_{2+}}{n_2}\right)^2-\left(\frac{n_{2-}}{n_2}\right)^2\right] \\ &= \frac{n_1+n_2}{N}-\frac{n_{1+}^2}{n_1N}-\frac{n_{1-}^2}{n_1N}-\frac{n_{2+}^2}{n_2N}-\frac{n_{2-}^2}{n_2N} \\ &= 1-\frac{n_{1+}^2}{n_1N}-\frac{n_{1-}^2}{n_1N}-\frac{n_{2+}^2}{n_2N}-\frac{n_{2-}^2}{n_2N} \end{aligned} \tag{3.2}$$

Comparing Equations (3.1) and (3.2), we can prove that gini index never increases after splitting (i.e., $g_1 \geq g_2$) by showing that:

$$\left(\frac{n_+}{N}\right)^2+\left(\frac{n_-}{N}\right)^2 \leq \frac{n_{1+}^2}{n_1N}+\frac{n_{2+}^2}{n_2N}+\frac{n_{1-}^2}{n_1N}+\frac{n_{2-}^2}{n_2N}$$

or,

$$\frac{n_+^2}{N} \leq \frac{n_{1+}^2}{n_1}+\frac{n_{2+}^2}{n_2} \quad \text{and} \tag{3.3}$$

$$\frac{n_-^2}{N} \leq \frac{n_{1-}^2}{n_1}+\frac{n_{2-}^2}{n_2} \tag{3.4}$$

To prove the inequality given in (3.3):

$$\begin{aligned} \frac{n_{1+}^2}{n_1}+\frac{n_{2+}^2}{n_2}-\frac{n_+^2}{N} &= \frac{n_{1+}^2}{n_1}+\frac{n_{2+}^2}{n_2}-\frac{(n_{1+}+n_{2+})^2}{n_1+n_2} \\ &= \frac{n_{1+}^2n_2(n_1+n_2)+n_{2+}^2n_1(n_1+n_2)-n_1n_2(n_{1+}+n_{2+})^2}{n_1n_2(n_1+n_2)} \\ &= \frac{n_{1+}^2n_2^2+n_{2+}^2n_1^2-2n_1n_2n_{1+}n_{2+}}{n_1n_2(n_1+n_2)} \\ &= \frac{(n_{1+}n_2-n_{2+}n_1)^2}{n_1n_2(n_1+n_2)} \\ &\geq 0 \end{aligned}$$

The last step follows from the fact that $(n_{1+}n_2 - n_{2+}n_1)^2 \geq 0$ (i.e., the square of any real numbers must be non-negative) and $n_1n_2(n_1+n_2) \geq 0$. Thus, the inequality in (3.3) holds. A similar proof can be given for the inequality in (3.4) by replacing n_{1+}, n_{2+}, and n_+ with n_{1-}, n_{2-}, and n_-, respectively.

4. Consider a data set that contains 4 Boolean attributes A, B, C, and D, and a Boolean class y. For each Boolean expression below (between the class y and the rest of the attributes), state whether it is possible to construct a smaller decision tree that perfectly classifies the data without generating the complete tree (i.e., the number of leave nodes is less than 16). If possible, draw the tree.

   (a) $y = A \wedge B \wedge C \wedge D$.

   Answer: Yes, it is possible to construct a smaller tree.

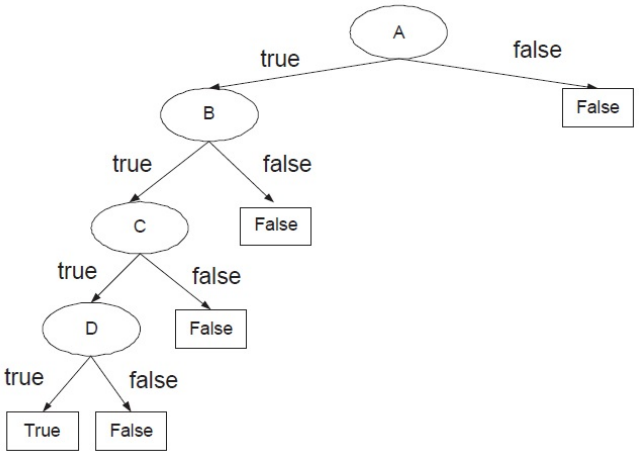


Figure 3.6. Decision tree for $y = A \wedge B \wedge C \wedge D$.

(b) $y = A \vee B \vee C \vee D$.

Answer: Yes, it is possible to construct a smaller tree.

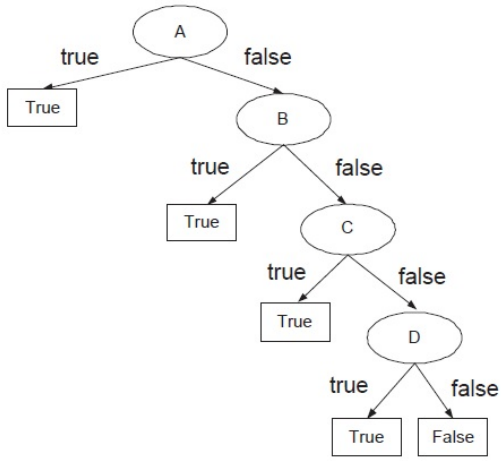


Figure 3.7. Decision tree for $y = A \vee B \vee C \vee D$.

(c) $y = (A \vee B) \wedge (C \vee D)$.

Answer: Yes, it is possible to construct a smaller tree.

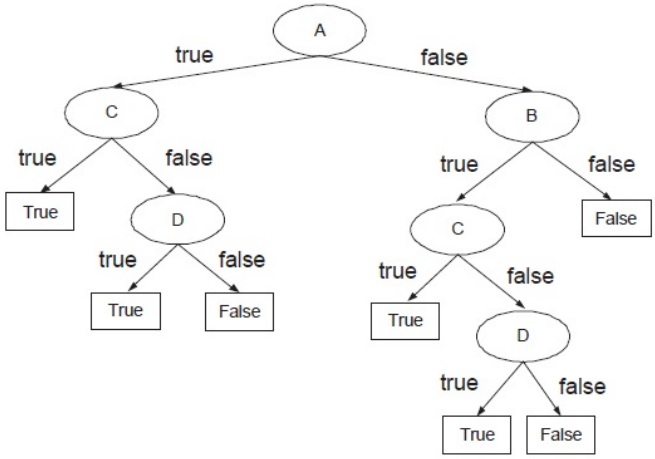


Figure 3.8. Decision tree for $y = (A \vee B) \wedge (C \vee D)$.

5. Consider the training set given below for predicting lung cancer in patients based on their symptoms (chronic cough and weight loss) and other lifestyle and environmental attributes (tobacco smoking and exposure to radon). Draw a two-level decision tree obtained using entropy as the impurity measure. Show your steps clearly (i.e., the computation of information gain for every candidate attribute at the first and second levels of the decision tree must be shown). Compute the training error of the decision tree.

Tobacco Smoking	Radon Exposure	Chronic Cough	Weight Loss	Lung Cancer
Yes	Yes	Yes	No	Yes
Yes	No	Yes	No	Yes
Yes	No	Yes	Yes	Yes
Yes	No	Yes	Yes	Yes
No	Yes	No	Yes	Yes
Yes	No	No	No	No
No	No	Yes	No	No
No	No	Yes	Yes	No
No	No	Yes	No	No
No	No	No	Yes	No

Answer:

Before splitting: $p(+) = p(-) = 0.5$. Therefore, the overall entropy is $-0.5\log(0.5) - 0.5\log(0.5) = 1$. The contingency tables and entropies after splitting on the attributes are:

		Lung Cancer		Entropy	Entropy	Information
		Yes	No	(Child)	Total	Gain
Tobacco Smoking	Yes	4	1	0.7219	0.7219	0.2781
	No	1	4	0.7219		
Radon Exposure	Yes	2	0	0	0.7635	0.2365
	No	3	5	0.9544		
Chronic Cough	Yes	4	3	0.9852	0.9651	0.0349
	No	1	2	0.9183		
Weight Loss	Yes	3	2	0.9710	0.9710	0.0290
	No	2	3	0.9710		

So, the attribute with highest information gain is tobacco smoking. Next, for tobacco smoking = yes, the contingency tables and entropies after splitting on the remaining attributes are:

Tobacco smoking = yes		Lung Cancer		Entropy (Child)	Entropy Total	Information Gain
		Yes	No			
Radon Exposure	Yes	1	0	0	0.6490	0.0729
	No	3	1	0.8113		
Chronic Cough	Yes	4	0	0	0	0.7219
	No	0	1	0		
Weight Loss	Yes	2	0	0	0.5510	0.1709
	No	2	1	0.9183		

Therefore, the best attribute to split the data (at level 2) for tobacco smoking = yes is Chronic cough. If chronic cough = yes, the leaf node is labeled as lung cancer = yes. If chronic cough = no, the leaf node is labeled as lung cancer = no.

For tobacco smoking = no, the contingency tables and entropies after splitting on the remaining attributes are:

Tobacco smoking = no		Lung Cancer		Entropy (Child)	Entropy Total	Information Gain
		Yes	No			
Radon Exposure	Yes	1	0	0	0	0.7219
	No	0	4	0		
Chronic Cough	Yes	0	3	0	0.4000	0.3219
	No	1	1	1		
Weight Loss	Yes	1	2	0.9183	0.5510	0.1709
	No	0	2	0		

Therefore, the best attribute to split the data (at level 2) for tobacco smoking = no is radon exposure. If radon exposure = yes, the leaf node is labeled as lung cancer = yes. If radon exposure = no, the leaf node is labeled as lung cancer = no. The 2-level decision tree is shown below and its training error is 0.

6. Show that the error rate of a decision tree never increases if one of the nodes is split into smaller successor nodes. To simplify the problem, you can assume that both the splitting attribute associated with the node and the class label are binary valued. It is sufficient to assume that the tree originally has only 1 node. After splitting, the new decision tree has 3 nodes (1 root node and 2 leave nodes). Show that the error rate of the new decision tree cannot be larger than the error rate of the initial tree.

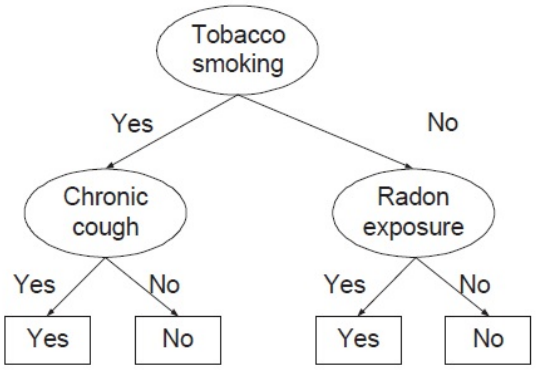


Figure 3.9. Decision tree for lung cancer prediction problem.

Answer: Consider a node (before splitting) with the following class distribution: $n_+ + n_- = N$, where n_+ and n_- are the number of training examples that belong to the positive and negative classes, respectively. The error rate of the node is:

$$\text{Error rate (before splitting)}, E_b = 1 - \max\left[\frac{n_+}{N}, \frac{n_-}{N}\right] \quad (3.5)$$

Suppose the node is split into two children. The class distribution for the first child is n_{1+} and n_{1-}, while the class distribution for the second child is n_{2+} and n_{2-}, respectively. The error rates of the children are

$$E_{c1} = 1 - \max\left[\frac{n_{1+}}{n_1}, \frac{n_{1-}}{n_1}\right], \quad E_{c2} = 1 - \max\left[\frac{n_{2+}}{n_2}, \frac{n_{2-}}{n_2}\right]$$

Therefore, the error rate after splitting is:

$$\begin{aligned} & \text{Error rate (after splitting)}, E_a = \frac{n_1}{N}E_{c1} + \frac{n_2}{N}E_{c2} \\ = & \frac{n_1}{N}\left[1 - \max\left(\frac{n_{1+}}{n_1}, \frac{n_{1-}}{n_1}\right)\right] + \frac{n_2}{N}\left[1 - \max\left(\frac{n_{2+}}{n_2}, \frac{n_{2-}}{n_2}\right)\right] \\ = & \frac{n_1 + n_2}{N} - \max\left(\frac{n_{1+}}{N}, \frac{n_{1-}}{N}\right) - \max\left(\frac{n_{2+}}{N}, \frac{n_{2-}}{N}\right) \\ = & 1 - \left[\max\left(\frac{n_{1+}}{N}, \frac{n_{1-}}{N}\right) + \max\left(\frac{n_{2+}}{N}, \frac{n_{2-}}{N}\right)\right] \end{aligned} \quad (3.6)$$

To complete the proof, we need to show that

$$\max\left(\frac{n_{1+}}{N}, \frac{n_{1-}}{N}\right) + \max\left(\frac{n_{2+}}{N}, \frac{n_{2-}}{N}\right) \geq \max\left[\frac{n_{+}}{N}, \frac{n_{-}}{N}\right]$$

Note that

$$\begin{aligned}
& \max\left(\frac{n_{1+}}{N}, \frac{n_{1-}}{N}\right) + \max\left(\frac{n_{2+}}{N}, \frac{n_{2-}}{N}\right) \\
= & \max\left(\frac{n_{1+}}{N} + \frac{n_{2+}}{N}, \frac{n_{1-}}{N} + \frac{n_{2-}}{N}, \frac{n_{1+}}{N} + \frac{n_{2-}}{N}, \frac{n_{1-}}{N} + \frac{n_{2+}}{N}\right) \\
= & \max\left(\frac{n_{+}}{N}, \frac{n_{-}}{N}, \frac{n_{1+}}{N} + \frac{n_{2-}}{N}, \frac{n_{1-}}{N} + \frac{n_{2+}}{N}\right) \\
\leq & \max\left(\frac{n_{+}}{N}, \frac{n_{-}}{N}\right) \qquad (3.7)
\end{aligned}$$

where, on the last line, we have used the following property of the max function, that adding any number to a list can only make the maximum value larger. Thus, $E_a \leq E_b$.

7. Consider the two-dimensional data shown in Figure 3.10(a). The data consists of two classes: A and B.

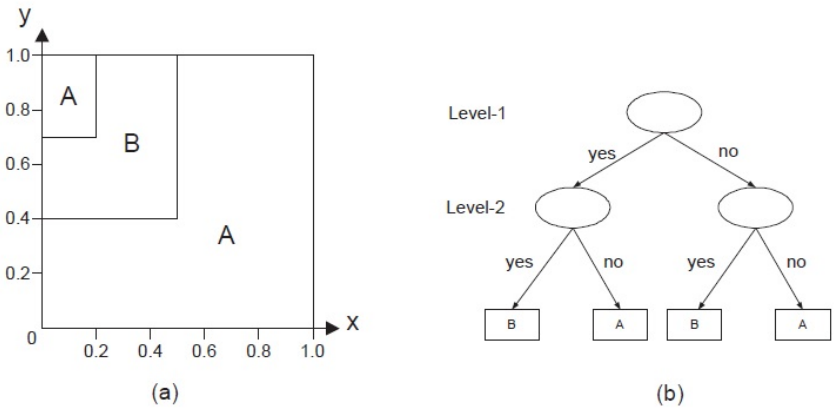


Figure 3.10. (a) A 2-d data set, (b) a 2-level decision tree.

(a) Draw a 2-level decision tree for the data (see Figure 3.10(b)). Use gini index as the splitting criterion. Assume the classifier uses a

binary split, i.e., the splitting criterion at each internal node must be specified either as $x \leq c$ or $y \leq c$, where c is some constant. In other words, do not specify the splitting criteria as $0.5 \leq x \leq 1.0$ or $x + y \leq 1$.

Answer: There are two important points to note here. First, the probabilities associated with each class is proportional to its area in the diagram. Second, the best split position is always located at the boundary between the two classes. Thus, the candidate split positions you need to consider at level 1 of the decision tree are $x \leq 0.2$, $x \leq 0.5$, $y \leq 0.7$, and $y \leq 0.4$.

		Class		Gini	Gini
		A	B	(Child)	Total
$x \leq 0.2$	Yes	0.14	0.06	0.4200	0.3630
	No	0.62	0.18	0.3488	
$x \leq 0.50$	Yes	0.26	0.24	0.4992	0.2496
	No	0.5	0	0	
$y \leq 0.4$	Yes	0.40	0	0	0.2880
	No	0.36	0.24	0.4800	
$y \leq 0.7$	Yes	0.55	0.15	0.3367	0.3617
	No	0.21	0.09	0.4200	

Thus, we should split at $x = 0.5$ because it has the lowest gini. Furthermore, for $x > 0.5$, notice that the entire region is classified as A, so it does not have to be split any further. For $x \leq 0.5$, we need to consider the following candidate split positions: $x \leq 0.2$, $y \leq 0.4$, and $y \leq 0.7$.

		Class		Gini	Gini
		A	B	(Child)	Total
$x \leq 0.2$	Yes	0.14	0.06	0.4200	0.4560
	No	0.12	0.18	0.4800	
$y \leq 0.4$	Yes	0.20	0	0	0.1920
	No	0.06	0.24	0.3200	
$y \leq 0.7$	Yes	0.20	0.15	0.4898	0.4869
	No	0.06	0.09	0.4800	

Clearly, the best split position is $y \leq 0.4$. If true, then the node is classified as A. Otherwise, it is classified as B. The resulting decision tree is shown in Figure 3.11.

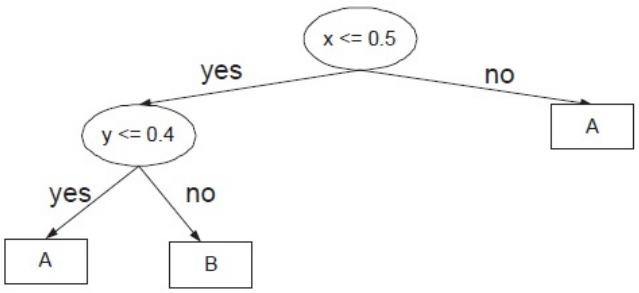


Figure 3.11. Decision tree for 2D problem.

(b) Compute the expected error rate of your decision tree when it is applied to a test set randomly sampled from the same 2-d space.

Answer: The expected error rate of the tree is equal to the area of the upper left-hand corner box labeled as A, which is equal to 0.06 or 6%.

8. Consider the decision trees shown in Figures 3.12(a) and (b). For each approach described below, you need to compute the generalization errors for both trees and decide which tree is better. The training and validation data sets are shown in Figures 3.12(c) and (d), respectively.

(a) Optimistic approach (assumes generalization error is given by the training error).

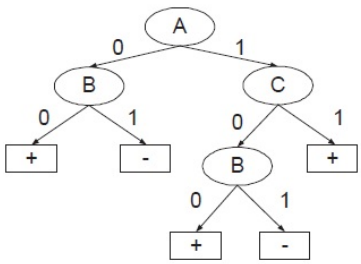
Answer: Error rates for trees A and B and 10% and 20%, respectively. So tree A is better.

(b) Pessimistic approach using the upper bound on generalization error with $\alpha = 0.25$ (or $Z_{1-\alpha/2} = 1.15$).

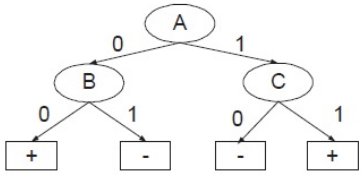
Answer: For this approach, you need to compute the expected error of each node and then add them up. Given a leaf node with N training examples that reach the node and an error rate of e, the upper bound on generalization error is:

$$e'(e, N) \leq \frac{e + \frac{z^2_{1-\alpha/2}}{2N} + z_{1-\alpha/2}\sqrt{\frac{e(1-e)}{N} + \frac{z^2_{1-\alpha/2}}{4N^2}}}{1 + \frac{z^2_{1-\alpha/2}}{N}}$$

For tree A, there are five leaf nodes. The upper bound on error rates of the nodes (going from left to right) are: $e_1(0, 2) = e_2(0, 2) =$



(a) Decision Tree A



(b) Decision Tree B

#	A	B	C	Class
1	0	0	0	+
2	0	0	0	+
3	0	1	0	-
4	0	1	1	-
5	1	0	0	+
6	1	0	1	+
7	1	0	1	-
8	1	1	0	-
9	1	1	0	-
10	1	1	1	+

(c) Training data

#	A	B	C	Class
11	0	0	0	+
12	0	0	1	-
13	0	1	0	-
14	0	1	1	-
15	0	1	1	+
16	1	0	0	-
17	1	0	0	-
18	1	0	1	-
19	1	1	0	-
20	1	1	1	+

(c) Validation data

Figure 3.12. Question 8.

0.3980, $e_3(0,1) = 0.5694$, $e_4(0,2) = 0.3980$, and $e_5(1/3,3) = 0.6500$. So, the total expected error for tree A is $2 \times 0.3980 + 2 \times 0.3980 + 1 \times 0.5694 + 2 \times 0.3980 + 3 \times 0.6500 = 4.9077$ and its expected error rate is $4.9077/10 = 0.4908$.

For tree B, there are only three leaf nodes. The upper bound on error rates of the nodes (going from left to right) are: $e_1(0,2) = e_2(0,2) = 0.3980$ and $e_3(1/3,3) = e_4(1/3,3) = 0.6500$. So, the total expected error for tree B is $2 \times 0.3980 + 2 \times 0.3980 + 3 \times 0.6500 + 3 \times 0.6500 = 5.4923$ and its expected error rate is $4.9077/10 = 0.5492$.

So, tree A is better.

(c) Reduced error pruning approach (generalization error is computed using the validation set shown in Figure 3.12(d)).

Answer: Error rates for trees A and B and 50% and 30%, respectively. So tree B is better.

(d) minimum description length (MDL) approach. The total description length of a tree is given by:

$$Cost(tree, data) = Cost(tree) + Cost(data|tree),$$

- Each internal node of the tree is encoded by the ID of the splitting attribute. If there are m attributes, the cost of encoding each attribute is $\log_2 m$ bits.
- Each leaf node is encoded using the ID of the class it is associated with. If there are k classes, the cost of encoding a class is $\log_2 k$ bits.
- $Cost(tree)$ is the cost of encoding all the nodes in the tree. To simplify the computation, you can assume that the total cost of the tree is obtained by adding up the costs of encoding each internal node and each leaf node.
- $Cost(data|tree)$ is encoded using the classification errors the tree commits on the training set. Each error is encoded by $\log_2 n$ bits, where n is the total number of training examples.

Answer:

Total description length for tree A is

$$4 \times \lceil \log_2 3 \rceil + 5 \times \lceil \log_2 2 \rceil + 1 \times \lceil \log_2 10 \rceil = 17 \text{ bits}$$

Total description length for tree B is

$$3 \times \lceil \log_2 3 \rceil + 4 \times \lceil \log_2 2 \rceil + 2 \times \lceil \log_2 10 \rceil = 18 \text{ bits}$$

So tree A is better.

9. Draw a decision tree that perfectly classifies each of the data sets described below. There could be more than one answer to each question (you only need to draw one tree). You do not have to create a sample of the data to answer this question. Assume there are no noise nor missing values.

   (a) Consider a data set with three Boolean attributes, A, B, and C, with a binary class label y whose value is positive if the number of attributes with True values is exceeds those with False values, and

negative otherwise. For example, if A=True, B=True, C=False, then y=+ (because there are more attributes with True values). Draw the full decision tree (with 8 leaf nodes) for the data. State whether it is possible to construct a smaller tree (with number of leaf nodes less than 8) that perfectly classifies the data. If possible, show the tree.

Answer: Full decision tree is shown in Figure 3.13

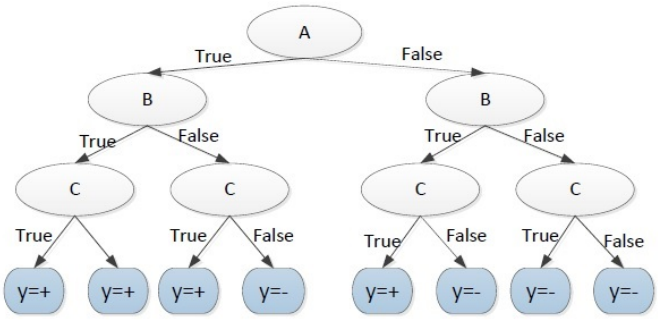


Figure 3.13. Full decision tree.

It is possible to construct a smaller tree, which is shown in Figure 3.14.

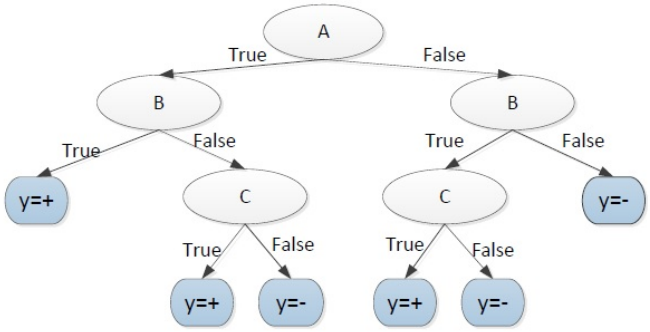


Figure 3.14. Pruned decision tree.

(b) Consider the diagram shown in Figure 3.15, where A, B, and C are the class labels associated with each region. Assuming a sufficiently

large number of training examples are sampled from each region (enough to learn the correct decision boundaries), draw a decision tree that would perfectly classify the data. You may assume the decision tree algorithm uses only binary splits (instead of multiway splits).

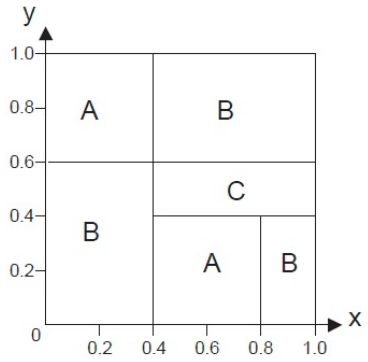


Figure 3.15. 2-D data set.

Answer: The decision tree is shown in Figure 3.16

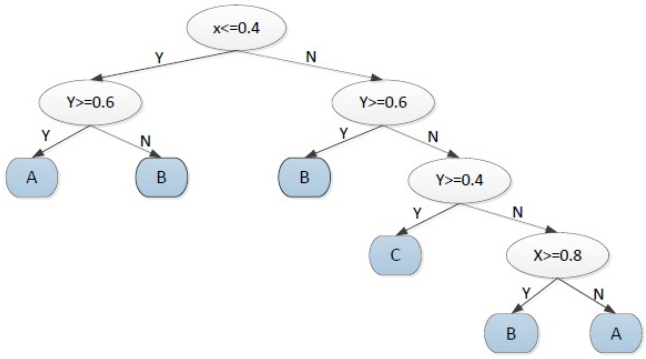


Figure 3.16. Decision tree for 2-D data set.

10. Consider the following data set that contains 100 training examples (50 labeled as positive class while the remainder labeled as negative class).

X	Y	Z	No. of + Examples	No. of − Examples
1	1	1	5	0
1	1	0	0	20
1	0	1	20	0
1	0	0	0	5
0	1	1	10	0
0	1	0	15	0
0	0	1	0	10
0	0	0	0	15

(a) Build a *two-level* decision tree using gini index as the criterion for splitting. You need to show your computations for each candidate splitting attribute at each level clearly to obtain full credit. What is the overall training error rate of the induced tree? Note: we consider a tree with only 1 internal node and two leaf nodes as a *one-level* decision tree.

Answer:

For level 1,

Before splitting: $Gini = 1-(\frac{5+20+10+15}{5+20+10+15+20+5+10+15})^2+(\frac{20+5+10+15}{5+20+10+15+20+5+10+15})^2 = 0.5$

If we split the node on attribute X,

	N_1	N_2
+	25	25
−	25	25

$$Gini(N_1) = 1 - (\frac{25}{50})^2 - (\frac{25}{50})^2 = 0.5$$

$$Gini(N_2) = 0.5$$

$$Gini(childern) = \frac{50}{100} \times 0.5 + \frac{50}{100} \times 0.5 = 0.5$$

If we split the node on attribute Y,

	N_1	N_2
+	30	20
−	20	30

$$Gini(N_1) = 1 - (\frac{30}{50})^2 - (\frac{20}{50})^2 = 0.48$$

$$Gini(N_2) = 1 - (\frac{20}{50})^2 - (\frac{30}{50})^2 = 0.48$$

$$Gini(childern) = \frac{50}{100} \times 0.48 + \frac{50}{100} \times 0.48 = 0.48$$

If we split the node on attribute Z,

	N_1	N_2
+	35	15
−	10	40

$$Gini(N_1) = 1 - (\frac{35}{45})^2 - (\frac{10}{45})^2 = 0.3457$$

$$Gini(N_2) = 1 - (\frac{15}{55})^2 - (\frac{40}{55})^2 = 0.3967$$

$$Gini(childern) = \frac{45}{100} \times 0.3457 + \frac{55}{100} \times 0.3967 = 0.3737$$

The $Gini(children)$ of Z is the smallest, hence we should split the node on Z. The first level tree is shown in Figure 3.17.

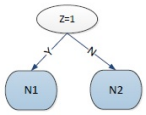


Figure 3.17. First split for 2(a).

For level 2:

For node N_1:

If we split the node on attribute X,

	N_1	N_2
+	25	10
−	0	10

$$Gini(N_1) = 1 - (\frac{25}{25})^2 - (\frac{0}{25})^2 = 0$$

$$Gini(N_2) = 1 - (\frac{10}{20})^2 - (\frac{10}{20})^2 = 0.5$$

$$Gini(childern) = \frac{25}{45} \times 0 + \frac{20}{45} \times 0.5 = 0.222$$

If we split the node on attribute Y,

	N_1	N_2
+	15	20
−	0	10

$$Gini(N_1) = 1 - (\frac{15}{15})^2 - (\frac{0}{15})^2 = 0$$

$$Gini(N_2) = 1 - (\frac{10}{30})^2 - (\frac{10}{30})^2 = 0.444$$

$$Gini(childern) = \frac{15}{45} \times 0 + \frac{30}{45} \times 0.444 = 0.296$$

The $Gini(children)$ of X is the smaller, hence we should split the node on X.

For node N_2:

If we split the node on attribute X,

	N_1	N_2
+	0	15
−	25	15

$$Gini(N_1) = 1 - (\frac{0}{25})^2 - (\frac{25}{25})^2 = 0$$

$$Gini(N_2) = 1 - (\frac{15}{30})^2 - (\frac{15}{30})^2 = 0.5$$

$$Gini(childern) = \frac{25}{55} \times 0 + \frac{30}{55} \times 0.5 = 0.273$$

If we split the node on attribute Y,

	N_1	N_2
+	15	0
−	20	20

$$Gini(N_1) = 1 - (\frac{15}{35})^2 - (\frac{20}{35})^2 = 0.4898$$

$$Gini(N_2) = 1 - (\frac{0}{20})^2 - (\frac{20}{20})^2 = 0$$

$$Gini(childern) = \frac{35}{55} \times 0.4898 + \frac{20}{55} \times 0 = 0.3117$$

The $Gini(children)$ of X is the smaller, hence we should split the node on X.

The two level decision tree is shown in Figure 3.18

The error rate is: $\frac{10+15}{100} = 0.25$.

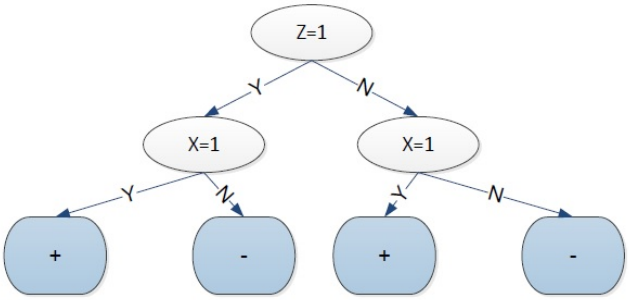


Figure 3.18. Two level tree for 2(a).

(b) Use variable X as the first splitting attribute, then choose the best available splitting attribute at each of the two successor nodes. What is the training error rate of the induced tree?

Answer:

If we split on X first,

For node N_1:

If we split the node on attribute Y,

	N_1	N_2
+	5	20
−	20	5

$$Gini(N_1) = 1 - (\frac{5}{25})^2 - (\frac{20}{25})^2 = 0.32$$

$$Gini(N_2) = 1 - (\frac{20}{25})^2 - (\frac{5}{25})^2 = 0.32$$

$$Gini(childern) = \frac{25}{50} \times 0.32 + \frac{25}{50} \times 0.32 = 0.32$$

If we split the node on attribute Z,

	N_1	N_2
+	25	0
−	0	25

$$Gini(N_1) = 1 - (\frac{25}{25})^2 - (\frac{0}{25})^2 = 0$$

$$Gini(N_2) = 1 - (\frac{0}{25})^2 - (\frac{25}{25})^2 = 0$$

$$Gini(childern) = \frac{25}{50} \times 0 + \frac{25}{50} \times 0 = 0$$

The $Gini(children)$ of Z is the smaller, hence we should split the node on Z.

For node N_2:

If we split the node on attribute Y,

	N_1	N_2
+	25	0
−	0	25

$$Gini(N_1) = 1 - (\frac{25}{25})^2 - (\frac{0}{25})^2 = 0$$

$$Gini(N_2) = 1 - (\frac{0}{25})^2 - (\frac{25}{25})^2 = 0$$

$$Gini(childern) = \frac{25}{50} \times 0 + \frac{25}{50} \times 0 = 0$$

If we split the node on attribute Y,

	N_1	N_2
+	10	15
−	10	15

$$Gini(N_1) = 1 - (\frac{10}{20})^2 - (\frac{10}{20})^2 = 0.5$$

$$Gini(N_2) = 1 - (\frac{15}{30})^2 - (\frac{15}{30})^2 = 0.5$$

$$Gini(childern) = \frac{20}{50} \times 0.5 + \frac{30}{50} \times 0.5 = 0.5$$

The $Gini(children)$ of Y is the smaller, hence we should split the node on Y.

The two level decision tree is shown in Figure 3.19

The error rate is: $\frac{10+15}{100} = 0$.

(c) Discuss the results obtained in parts (a) and (b) above. Comment on the suitability of the greedy heuristic used as the splitting attribute selection.

Answer: Greedy heuristic method cannot guarantee to produce the optimal decision tree.

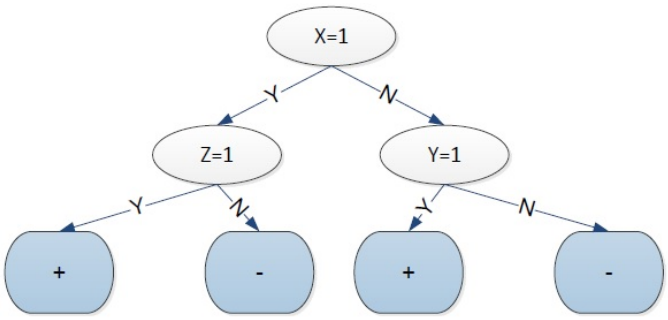


Figure 3.19. Two level tree for 2(a).

11. Consider the problem of predicting how well a baseball player will bat against a particular pitcher. The training set contains ten positive and ten negative examples. Assume there are two candidate attributes for splitting the data—ID (which is unique for every player) and Handedness (left or right). Among the left-handed players, nine of them are from the positive class and one from the negative class. On the other hand, among the right-handed players, only one of them is from the positive class, while the remaining nine are from the negative class.

    (a) Compute the information gain if we use ID as the splitting attribute.
    Answer: The entropy for the parent node is:

    $$Entropy_{parent} = -(0.5 \log 0.5 + 0.5 \log 0.5) = 1$$

    If split using ID,

    $$Gain = Entropy_{parent} - (\sum_{i=1}^{20} Entropy(i)) = 1 - 20 \times \frac{1}{20}(-0 \log 0 - 1 \log 1) = 1$$

    (b) Repeat part (a) using Handedness as the splitting attribute.
    Answer: If split using ID,

    $$Gain = Entropy_{parent} - \frac{10}{20}(-0.9 \log 0.9 - 0.1 \log 0.1) \times 2 = 1 - 0.469 = 0.531$$

    (c) Based on your answers in parts (a) and (b), which attribute will be chosen according to information gain?
    Answer: According to information gain, we should choose ID to split the node.

(d) Repeat part (a) using gain ratio (instead of information gain).

Answer:

$GainRATIO_{split} = \frac{Gain}{SplitInfo} = \frac{1}{20 \times (-0.05 \log 0.05)} = \frac{1}{4.322} = 0.231$

(e) Repeat part (b) using gain ratio (instead of information gain).

Answer:

$GainRATIO_{split} = \frac{Gain}{SplitInfo} = \frac{0.531}{2 \times (-0.5 \log 0.5)} = \frac{0.531}{1} = 0.531$

(f) Based on your answers in parts (d) and (e), which attribute will be chosen according to gain ratio?

Answer: According to the gain ratio, we should choose Handedness to split the node.

12. Consider the training set given below for determining whether a loan application should be approved or rejected. Draw the full decision tree obtained using entropy as the impurity measure. Show your steps clearly (i.e., the computation of information gain for every candidate attribute must be shown). Compute the training error of the decision tree.

Long-Term Debt	Unemployed	Credit Rating	Down Payment $< 20\%$	Class
No	No	Good	Yes	Approve
No	No	Bad	No	Approve
No	No	Bad	Yes	Approve
No	No	Bad	No	Approve
Yes	No	Good	No	Approve
No	Yes	Good	Yes	Reject
Yes	No	Bad	No	Reject
Yes	No	Bad	Yes	Reject
Yes	No	Bad	Yes	Reject
Yes	Yes	Bad	No	Reject

Answer:

Before splitting: $p(+) = p(-) = 0.5$. Therefore, the overall entropy is $-0.5 \log(0.5) - 0.5 \log(0.5) = 1$. The contingency tables and entropies after splitting on the attributes are:

		Class		Entropy	Entropy	Info
		Reject	Approve	(Child)	Total	Gain
Long-Term Debt	Yes	4	1	0.7219	0.7219	0.2781
	No	1	4	0.7219		
Unemployed	Yes	2	0	0	0.7635	0.2365
	No	3	5	0.9544		
Credit Rating	Bad	4	3	0.9852	0.9651	0.0349
	Good	1	2	0.9183		
Down Payment $< 20\%$	Yes	3	2	0.9710	0.9710	0.0290
	No	2	3	0.9710		

So, the attribute with highest information gain is long-term debt. Next, for long-term debt = yes, the contingency tables and entropies after splitting on the remaining attributes are:

Long-term debt = yes		Class		Entropy	Entropy	Info
		Reject	Approve	(Child)	Total	Gain
Unemployed	Yes	1	0	0	0.6490	0.0729
	No	3	1	0.8113		
Credit Rating	Bad	4	0	0	0	0.7219
	Good	0	1	0		
Down Payment $< 20\%$	Yes	2	0	0	0.5510	0.1709
	No	2	1	0.9183		

Therefore, the best attribute to split the data (at level 2) for long-term debt = yes is credit rating. If credit rating = bad, the leaf node is labeled as class = reject. If credit rating = good, the leaf node is labeled as class = approve.

For long-term debt = no, the contingency tables and entropies after splitting on the remaining attributes are:

Long-term debt = no		Class		Entropy (Child)	Entropy Total	Info Gain
		Reject	Approve			
Unemployed	Yes	1	0	0	0	0.7219
	No	0	4	0		
Credit Rating	Bad	0	3	0	0.4000	0.3219
	Good	1	1	1		
Down Payment $< 20\%$	Yes	1	2	0.9183	0.5510	0.1709
	No	0	2	0		

Therefore, the best attribute to split the data (at level 2) for long-term debt = no is unemployed. If unemployed = yes, the leaf node is labeled as class = reject. If unemployed = no, the leaf node is labeled as class = approve. The decision tree is shown below and its training error is 0.

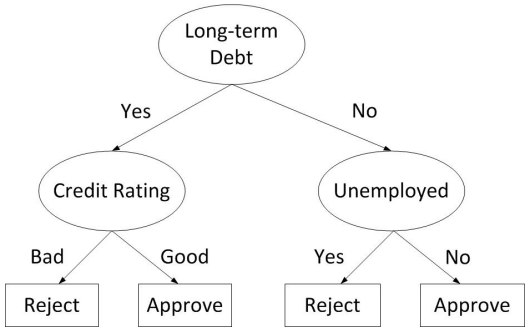


Figure 3.20. Decision tree for loan approval problem.

13. This question examines property of the entropy measure.

    (a) Show that the entropy measure $-p(x)\log p(x)$ is non-negative.
    Answer: Since $0 \leq p(x) \leq 1$, hence $\log p(x) \leq 0$. Hence, $-p(x)\log p(x) \geq 0$.

    (b) Consider a pair of binary variables, X and Y. Suppose we need to estimate their joint probability distribution, $P(X, Y)$ as shown in the following 2×2 table:

	$Y = 1$	$Y = 0$
$X = 1$	$P(X = 1, Y = 1) = a$	$P(X = 1, Y = 0) = b$
$X = 0$	$P(X = 0, Y = 1) = c$	$P(X = 0, Y = 0) = d$

Find the joint probabilities (i.e., values of a, b, c, and d) that will maximize the entropy of the distribution assuming we know that $P(X = 1) = a + b = 0.7$ and $\sum_{X,Y} P(X, Y) = a + b + c + d = 1$. Hint: solve the constraint optimization problem where the objective function corresponds to the total entropy of the distribution (refer to lecture 3 on how to solve a constraint optimization problem with the Lagrange multiplier method).

Answer:

The entropy of the distribution is:

$$Entropy = -(a \log a + b \log b + c \log c + d \log d).$$

$L = -(a \log a + b \log b + c \log c + d \log d) - \lambda_1(a+b+c+d-1) - \lambda_2(a+b-0.7)$

Take derivative with respect to a,b,c,d,λ_1,λ_2, we get:

$$\frac{\partial L}{\partial a} = \log a + \frac{1}{ln2} - \lambda_1 - \lambda_2 = 0$$

$$\frac{\partial L}{\partial b} = \log b + \frac{1}{ln2} - \lambda_1 - \lambda_2 = 0$$

$$\frac{\partial L}{\partial c} = \log c + \frac{1}{ln2} - \lambda_1 = 0$$

$$\frac{\partial L}{\partial d} = \log d + \frac{1}{ln2} - \lambda_1 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = -(a + b + c + d - 1) = 0$$

$$\frac{\partial L}{\partial \lambda_2} = -(a + b - 0.7) = 0$$

The solutions are: $a = b = 0.35$, $c = d = 0.15$.

(c) Consider a nominal attribute X that has three possible values, x_1, x_2, and x_3. Suppose you have a decision tree classifier that can produce either a multi-way split or a binary split on attribute X. Show that the average entropy of the successors for node X in a multi-way split is always smaller than or equal to the average entropy of the successors of node X in a binary split. Hint: you can apply the following Gibbs inequality for a given pair of probability distributions, p and q, for the proof:

$$-\sum_i p_i \log p_i \leq \sum_i p_i \log q_i.$$

Answer:

Assume this is a binary classifier, and the classes are labeled as "+" and "−". The numbers of samples in the nodes split by x_i is denoted by n_i in the multi-way split and in the binary split the numbers of samples are denoted as n'_1 and $n'_{2,3}$. N is the total number of samples. It is obvious that $n'_1 = n_1$ and $n_{2,3} = n_2 + n_3$. "n_i^+" and "n_i^-" are represented as the number of samples belonging to class "+" and class "+" respectively in node n_i.

For multi-way split, the entropy is:

$$\begin{aligned} Entropy_1 = & - \frac{n_1}{N}\left(\frac{n_1^+}{n_1}\log\frac{n_1^+}{n_1} + \frac{n_1^-}{n_1}\log\frac{n_1^-}{n_1}\right) \\ & - \frac{n_2}{N}\left(\frac{n_2^+}{n_2}\log\frac{n_2^+}{n_2} + \frac{n_2^-}{n_2}\log\frac{n_2^-}{n_2}\right) \\ & - \frac{n_3}{N}\left(\frac{n_3^+}{n_3}\log\frac{n_3^+}{n_3} + \frac{n_3^-}{n_3}\log\frac{n_3^-}{n_3}\right) \end{aligned}$$

For binary split, the entropy is:

$$\begin{aligned} Entropy_2 = & - \frac{n_1}{N}\left(\frac{n_1^+}{n_1}\log\frac{n_1^+}{n_1} + \frac{n_1^-}{n_1}\log\frac{n_1^-}{n_1}\right) \\ & - \frac{n_2+n_3}{N}\left(\frac{n_2^+ + n_3^+}{n_2+n_3}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_2^- + n_3^-}{n_2+n_3}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right) \end{aligned}$$

If we want to compare the above two equations, we only need to compare these two equations:

$$E_1 = -\frac{n_2}{N}\left(\frac{n_2^+}{n_2}\log\frac{n_2^+}{n_2} + \frac{n_2^-}{n_2}\log\frac{n_2^-}{n_2}\right) - \frac{n_3}{N}\left(\frac{n_3^+}{n_3}\log\frac{n_3^+}{n_3} + \frac{n_3^-}{n_3}\log\frac{n_3^-}{n_3}\right)$$

and

$$E_2 = -\frac{n_2+n_3}{N}\left(\frac{n_2^+ + n_3^+}{n_2+n_3}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_2^- + n_3^-}{n_2+n_3}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right)$$

Using Gibbs' inequality, we can show that,

$$
\begin{aligned}
E_1 &= -\frac{n_2}{N}\left(\frac{n_2^+}{n_2}\log\frac{n_2^+}{n_2} + \frac{n_2^-}{n_2}\log\frac{n_2^-}{n_2}\right) - \frac{n_3}{N}\left(\frac{n_3^+}{n_3}\log\frac{n_3^+}{n_3} + \frac{n_3^-}{n_3}\log\frac{n_3^-}{n_3}\right) \\
&\leq -\frac{n_2}{N}\left(\frac{n_2^+}{n_2}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_2^-}{n_2}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right) \\
&\quad -\frac{n_3}{N}\left(\frac{n_3^+}{n_3}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_3^-}{n_3}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right) \\
&= -\left(\frac{n_2^+ + n_3^+}{N}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_2^- + n_3^-}{N}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right) \\
&= -\frac{n_2+n_3}{N}\left(\frac{n_2^+ + n_3^+}{n_2+n_3}\log\frac{n_2^+ + n_3^+}{n_2+n_3} + \frac{n_2^- + n_3^-}{n_2+n_3}\log\frac{n_2^- + n_3^-}{n_2+n_3}\right) \\
&= E_2
\end{aligned}
$$

Hence, $Entropy_1 \leq Entropy_2$.

14. This question examines the relationship between entropy (H) and mutual information (I).

    (a) Based on the following definitions:

$$
\begin{aligned}
I(X,Y) &= \sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
H(X) &= -\sum_{x\in X} p(x)\log p(x) \\
H(X,Y) &= -\sum_{x\in X, y\in Y} p(x,y)\log p(x,y)
\end{aligned}
$$

    Prove that $I(X,Y) = H(X) + H(Y) - H(X,Y)$.

Answer:

$$
\begin{aligned}
I(X,Y) &= \sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
&= \sum_{x\in X, y\in Y} p(x,y)\log p(x,y) - \sum_{x\in X, y\in Y} p(x,y)\log p(x) \\
&\quad - \sum_{x\in X, y\in Y} p(x,y)\log p(y) \\
&= \sum_{x\in X, y\in Y} p(x,y)\log p(x,y) - \sum_{x\in X}\left(\sum_{y\in Y} p(x,y)\right)\log p(x) \\
&\quad - \sum_{y\in Y}\left(\sum_{x\in X} p(x,y)\right)\log p(y) \\
&= \sum_{x\in X, y\in Y} p(x,y)\log p(x,y) - \sum_{x\in X} p(x)\log p(x) \\
&\quad - \sum_{y\in Y} p(y)\log p(y) \\
&= -H(X,Y) + H(X) + H(Y)
\end{aligned}
$$

(b) Consider a data source that generates a letter α from a set of alphabets $\{a, b, c, \cdots, z\}$. If a vowel (a, e, i, o, u) is four times more likely to be generated than a consonant $(b, c, d, f, \cdots, z)$, calculate the entropy for α.

Answer: Let V be the set of vowels and C be the set of consonants. Based on the given information:

$$
P(\alpha = \alpha_i) = \begin{cases} 4p, & \text{if } \alpha_i \in \text{V}; \\ p, & \text{if } \alpha_i \in \text{C}. \end{cases} \tag{3.8}
$$

Since there are 5 vowels and 21 consonants and $\sum_{i=1}^{26} P(\alpha = \alpha_i) = 1$:

$$
5 \times 4p + 21 \times p = 1 \implies p = 1/41.
$$

Therefore, entropy for the random variable α is

$$\begin{aligned}\text{Entropy}(\alpha) &= -\sum_{i=1}^{26} P(\alpha_i)\log_2 P(\alpha_i) \\ &= -5 \times \frac{4}{41}\log_2\frac{4}{41} - 21 \times \frac{1}{41}\log_2\frac{1}{41} \\ &= 4.3819 \text{ bits.}\end{aligned}$$

(c) Consider a pair of binary variables, X and Y. Suppose we need to estimate their joint probability distribution, $P(X, Y)$ as shown in the following 2×2 table:

	$Y = 1$	$Y = 0$
$X = 1$	$P(X = 1, Y = 1) = a$	$P(X = 1, Y = 0) = b$
$X = 0$	$P(X = 0, Y = 1) = c$	$P(X = 0, Y = 0) = d$

Find the joint probabilities (i.e., values of a, b, c, and d) that will maximize the entropy of the distribution assuming we know that $P(X = 1) = a + b = 0.6$ and $\sum_{X,Y} P(X, Y) = a + b + c + d = 1$. Hint: solve the constraint optimization problem where the objective function corresponds to the total entropy of the distribution.

Answer: We can pose this as the following optimization problem:

$$\begin{aligned}\max_{a,b,c,d} \quad & -a\log a - b\log b - c\log c - d\log d \\ s.t. \quad & a + b = 0.6 \\ & a + b + c + d = 1\end{aligned}$$

By using the Lagrange multiplier method, define the Lagrangian as:

$$L = -a\log a - b\log b - c\log c - d\log d - \lambda(a + b - 0.6) - \mu(a + b + c + d - 1)$$

Take the partial derivatives with respect to a, b, c, and d, and set them to zero:

$$\begin{aligned}
\frac{\partial L}{\partial a} &= -\log_2 a - \frac{1}{\ln 2} - \lambda - \mu = 0 \\
\frac{\partial L}{\partial b} &= -\log_2 b - \frac{1}{\ln 2} - \lambda - \mu = 0 \\
\frac{\partial L}{\partial c} &= -\log_2 c - \frac{1}{\ln 2} - \mu = 0 \\
\frac{\partial L}{\partial d} &= -\log_2 d - \frac{1}{\ln 2} - \mu = 0
\end{aligned}$$

Therefore,

$$\begin{aligned}
a = b &= 2^{-(1/\ln 2 + \lambda + \mu)} \\
c = d &= 2^{-(1/\ln 2 + \mu)}
\end{aligned}$$

Since $a + b = 0.6$, this reduces to $a = b = 0.3$. Plugging this into $a + b + c + d = 1$ and using $c = d$, this gives $c = d = 0.2$.

(d) Based on your answer in part (c), calculate the mutual information between X and Y.

Answer:

$$\begin{aligned}
&P(X = 1) = a + b = 0.6, P(X = 0) = c + d = 0.4 \\
&P(Y = 1) = a + c = 0.5, P(Y = 1) = b + d = 0.5 \\
&H(X) = -0.6 \log 0.6 - 0.4 \log 0.4 = 0.971 \\
&H(Y) = -0.5 \log 0.5 - 0.5 \log 0.5 = 1.000 \\
&H(X, Y) = -2 \times 0.3 \log 0.3 - 2 \times 0.2 \log 0.2 = 1.971
\end{aligned}$$

Therefore, mutual information is $H(X) + H(Y) - H(X, Y) = 0$.

15. This questions compares entropy against Gini index as impurity measures for decision trees.

(a) Consider a two-class problem. Show that the entropy of a node in the decision tree is always greater than or equal to its gini index (use $\log_2$ for entropy).

Answer: Let p and $(1-p)$ be the distribution of the two classes at a given node. The entropy of the node is $E = -p\log_2 p - (1-p)\log_2(1-p)$ whereas its gini is $G = 1 - p^2 - (1-p)^2$. We need to show that

$$E - G = -p\log_2 p - (1-p)\log_2(1-p) - 1 + p^2 + (1-p)^2 \geq 0$$

First, note that the negative logarithm is a convex function. According to Jensen inequality, for any convex function $f(x)$:

$$\sum_i \alpha_i f(x_i) \geq f(\sum_i \alpha_i x_i),$$

where $\sum_i \alpha_i = 1$. Thus:

$$\begin{aligned} E - G &= -p\log_2 p - (1-p)\log_2(1-p) - 1 + p^2 + (1-p)^2 \\ &= -p\log_2 p - (1-p)\log_2(1-p) - 2p(1-p) \\ &\geq -\log_2\Big(p^2 + (1-p)^2\Big) - 2p(1-p) \\ &= -\log_2\Big(1 - 2p(1-p)\Big) - 2p(1-p) \end{aligned} \tag{3.9}$$

where Jensen inequality was applied on the third line. Based on the Taylor series expansion of the logarithm function:

$$\log(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n}$$

Thus

$$\begin{aligned} E - G &\geq \sum_{n=1}^{\infty} \frac{\Big(2p(1-p)\Big)^n}{n} - 2p(1-p) \\ &= \sum_{n=2}^{\infty} \frac{\Big(2p(1-p)\Big)^n}{n} \end{aligned}$$

which is always non-negative since $2p(1-p) \geq 0$ when $0 \leq p \leq 1$. The latter can be proved as follows. By contradiction, suppose

$2p(1-p) < 0$ and $0 \leq p \leq 1$. The second inequality implies $p \geq 0$ and $(1-p) \geq 0$. But, since $2p(1-p) < 0$, either $p < 0$ or $(1-p) < 0$, which is a contradiction. Thus, $2p(1-p)$ must be non-negative.

(b) Consider a decision tree classifier for two-class problem. Suppose we have the option of choosing either the binary attribute **A** or **B** as our splitting condition (see Figure 3.21). If attribute **A** is preferred over attribute **B** as the splitting condition according to the entropy measure, is it possible for attribute **B** to be preferred over attribute **A** according to gini index? If so, give an example; otherwise, prove that it is impossible.

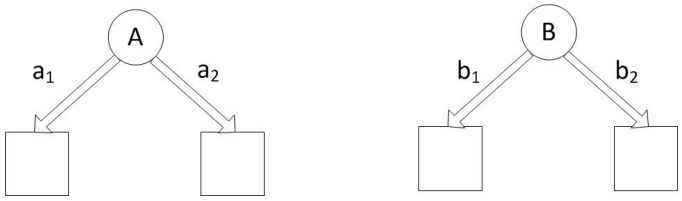


Figure 3.21. Two options for decision tree

Answer: Yes, it is possible for the different measures to select different attributes as the splitting condition. Consider the following class distribution of the child nodes for nodes A and B. Assume there are 4 positives and 6 negative examples in the training data (before splitting). The contingency tables after splitting on attributes A and B are:

	$A = a_1$	$A = a_2$
+	4	0
−	3	3

	$B = b_1$	$B = b_2$
+	3	1
−	1	5

The weighted entropy of the children after splitting on A is:

$$\begin{aligned} E_{A=a_1} &= -\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7} = 0.9852 \\ E_{A=a_2} &= -\frac{3}{3}\log\frac{3}{3} - \frac{0}{3}\log\frac{0}{3} = 0 \\ E_A &= 7/10 E_{A=a_1} + 3/10 E_{A=a_2} = 0.6897 \end{aligned}$$

The weighted entropy of the children after splitting on B is:

$$\begin{aligned} E_{B=b_1} &= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = 0.8113 \\ E_{B=b_2} &= -\frac{1}{6}\log\frac{1}{6} - \frac{5}{6}\log\frac{5}{6} = 0.6500 \\ E+B &= 4/10 E_{B=b_1} + 6/10 E_{B=b_2} = 0.7145 \end{aligned}$$

Since $E_A < E_B$, attribute A will be chosen to split the node.
The gini after splitting on A is:

$$\begin{aligned} G_{A=a_1} &= 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.4898 \\ G_{A=a_2} &= 1 = \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0 \\ G_A &= 7/10 G_{A=a_2} + 3/10 G_{A=a_2} = 0.3429 \end{aligned}$$

The gini after splitting on B is:

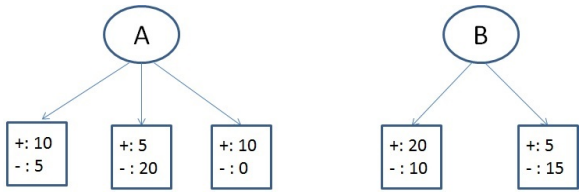
$$\begin{aligned} G_{B=b_1} &= 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.3750 \\ G_{B=b_2} &= 1 = \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.2778 \\ G_B &= 4/10 G_{B=b_1} + 6/10 G_{B=b_2} = 0.3167 \end{aligned}$$

Since $G_B < G_A$, attribute B will be chosen to split the node.

16. Consider a node in a decision tree with n_+ positive and n_- negative training examples. If the node is split to its k children, The average weighted entropy of the children is given by

$$\text{Entropy(children)} = \sum_k \frac{n_k}{n} \text{Entropy}(t_k),$$

where n_k is the number of training examples associated with the child node t_k and $n = \sum_k n_k = n_+ + n_-$. Apply the formula to calculate the average weighted entropy for each of the candidate test conditions shown below. Based on their entropy values, which attribute, A or B, should be chosen to split the parent node?



Answer: For node A, the entropies of its children are

$$\begin{aligned}
\text{Entropy(left)} &= -\frac{10}{15}\log_2\frac{10}{15} - \frac{5}{15}\log_2\frac{5}{15} = 0.9183 \\
\text{Entropy(middle)} &= -\frac{5}{25}\log_2\frac{5}{25} - \frac{20}{25}\log_2\frac{20}{25} = 0.7219 \\
\text{Entropy(right)} &= -\frac{10}{10}\log_2\frac{10}{10} - \frac{0}{10}\log_2\frac{0}{10} = 0
\end{aligned}$$

The average weighted entropy for node A is

$$\frac{15}{50} \times 0.9183 + \frac{25}{50} \times 0.7219 + \frac{10}{50} \times 0 = 0.6365$$

Similarly, for node B, the entropies of its children are

$$\begin{aligned}
\text{Entropy(left)} &= -\frac{20}{30}\log_2\frac{20}{30} - \frac{10}{30}\log_2\frac{10}{30} = 0.9183 \\
\text{Entropy(right)} &= -\frac{5}{20}\log_2\frac{5}{20} - \frac{15}{20}\log_2\frac{15}{20} = 0.8113
\end{aligned}$$

The average weighted entropy for node B is

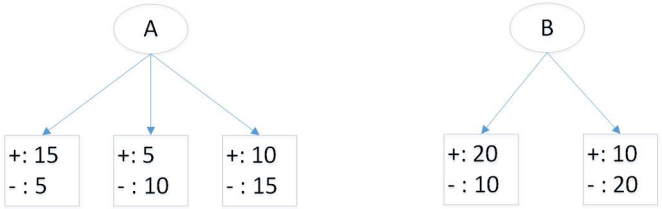
$$\frac{30}{50} \times 0.9183 + \frac{20}{50} \times 0.8113 = 0.8755$$

Based on their average weighted entropy values, node A should be chosen because its value is smaller.

17. Consider a node in a decision tree with n_+ positive and n_- negative training examples. If the node is split to its k children, The average weighted gini index of the children is given by

$$\text{Gini(children)} = \sum_k \frac{n_k}{n}\text{Gini}(t_k),$$

where n_k is the number of training examples associated with the child node t_k and $n = \sum_k n_k = n_+ + n_-$. Apply the formula to calculate the average weighted gini for each of the candidate test conditions shown below. Based on their gini values, which attribute, A or B, should be chosen to split the parent node?



Answer:

After splitting on attribute A, the gini index for each child is as follows:

$$\begin{aligned}
\text{Gini(left)} &= 1 - (\frac{15}{20})^2 - (\frac{5}{20})^2 = 0.3750 \\
\text{Gini(middle)} &= 1 - (\frac{5}{15})^2 - (\frac{10}{15})^2 = 0.4444 \\
\text{Gini(right)} &= 1 - (\frac{10}{25})^2 - (\frac{15}{25})^2 = 0.4800
\end{aligned}$$

Thus, the average weighted gini for the children is

$$\text{Gini(children)} = \frac{20}{60} \times 0.3750 + \frac{15}{60} \times 0.4444 + \frac{25}{60} \times 0.4800 = 0.4361.$$

After splitting on attribute B, the gini index for each child is as follows:

$$\begin{aligned}
\text{Gini(left)} &= 1 - (\frac{20}{30})^2 - (\frac{10}{30})^2 = 0.4444 \\
\text{Gini(right)} &= 1 - (\frac{10}{30})^2 - (\frac{20}{30})^2 = 0.4444
\end{aligned}$$

Thus, the average weighted gini for the children is

$$\text{Gini(children)} = \frac{30}{60} \times 0.4444 + \frac{30}{60} \times 0.4444 = 0.4444.$$

Based on the results, attribute A should be chosen to split the data since it has a lower gini.

18. Consider the decision tree shown in Figure 3.22 for a binary classification problem. Assume the classes are denoted as + and −, respectively. Suppose there are four binary attributes in the data, A, B, C, and D. The counts shown in the leaf nodes of the tree correspond to the number of training examples assigned to the nodes. Assume that the decision tree classifier assigns the majority class of training examples as the class label of each leaf node.

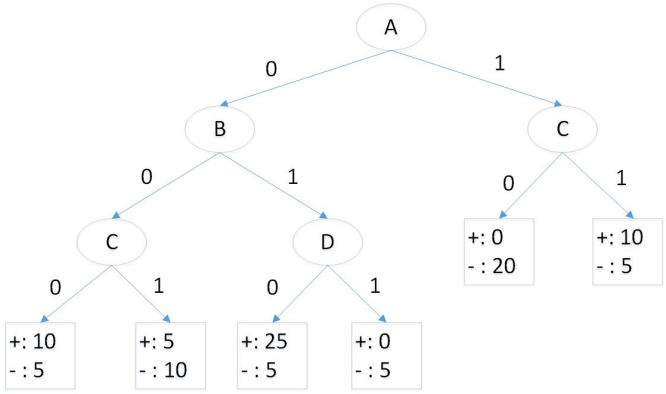


Figure 3.22. Unpruned decision tree

(a) Calculate the training error rate of the decision tree.

Answer: There are 100 examples in the training set. Assuming the leaf nodes are assigned to the majority class of training examples, the training error rate of the tree is $\frac{20}{100} = 0.2$.

(b) Calculate the generalization error rate of the decision tree using the validation set given below. Note that the wildcard $*$ shown in the table means the value could be either 0 or 1.

A	B	C	D	Number of + examples	Number of − examples
0	0	0	*	20	10
0	0	1	*	0	5
0	1	*	0	10	5
0	1	*	1	5	5
1	*	0	*	5	25
1	*	1	*	10	0

Answer: There are 100 examples in the validation set. The estimated generalization error of the tree is $\frac{25}{100} = 0.25$.

(c) Calculate the training and generalization error estimate of the pruned decision tree shown in Figure 3.23. Based on your estimate of gen-

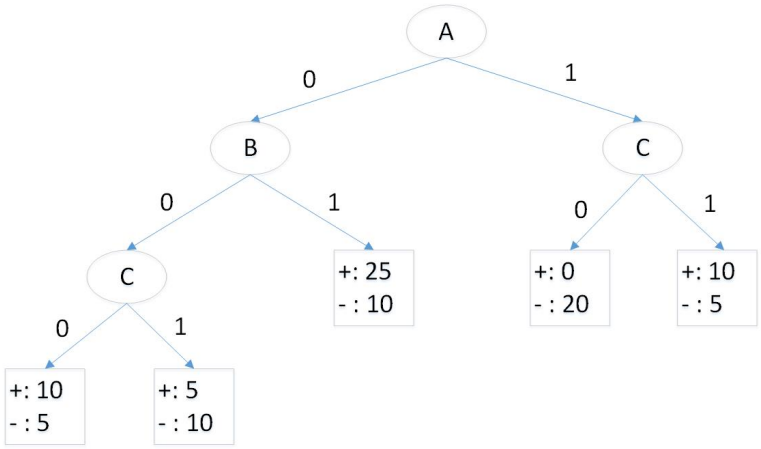


Figure 3.23. Pruned decision tree

eralization error, which tree should be preferred (the unpruned tree given in Figure 3.22 or the pruned tree given in Figure 3.23)?

Answer: The training error of the pruned tree is $\frac{25}{100} = 0.25$ while its generalization error is $\frac{25}{100} = 0.25$. According to estimated generalization errors using the validation set, both trees are equivalent. However, since the pruned tree is simpler, it should be preferred over the unpruned tree.

(d) Apply the minimum description length principle to determine which tree should be preferred (the unpruned tree given in Figure 3.22 or the pruned tree given in Figure 3.23). Assume the tree requires $\log_2 c$ bits to encode each leaf node (where c is number of classes), $\log_2 d$ bits to encode each internal node (where d is number of attributes), and $\log_2 N$ bits to encode each misclassified training example (where N is the number of training examples).

Answer: The unpruned tree has 5 internal and 6 leaf nodes. It also made 20 mistakes on a training set of size 100. Thus, the total

description length for the unpruned tree is

$$6 \times \log_2 2 + 5 \times \log_2 4 + 20 \times \lceil \log_2 100 \rceil = 156 \text{ bits}$$

The pruned tree has 4 internal and 5 leaf nodes. It made 25 mistakes on the training set. So, the total description length for the pruned tree is

$$5 \times \log_2 2 + 4 \times \log_2 4 + 25 \times \lceil \log_2 100 \rceil = 188 \text{ bits}$$

According to the MDL principle, the unpruned tree should be preferred over the pruned one.

3.2 Model Evaluation

1. You have been asked to develop a classification model for diagnosing whether a patient is infected with a certain disease. To help you construct the models, your collaborator has provided you with a small training set ($N = 10$) with equal number of positive and negative examples. You tried several approaches and found two most promising models, C_1 and C_2. The outputs of the models in terms of predicting whether each of the training examples belong to the "positive" class are summarized in the table below. The first row shows the probability a training example belongs to the positive class according to classifier C_1, while the second row shows the same information for classifier C_2. The last row indicates the true class label of the 10 training examples.

$P(y = +\|C_1)$	0.1	0.15	0.2	0.3	0.31	0.4	0.62	0.77	0.81	0.95
$P(y = +\|C_2)$	0.25	0.49	0.05	0.35	0.66	0.6	0.7	0.65	0.55	0.99
y	-	+	-	-	+	-	+	+	-	+

   (a) Draw the corresponding ROC curves for both classifiers on the same plot.

   Answer: See Figure 3.24.

   (b) Compute the area under ROC curve for each classifier. Which classifier has a larger area under the ROC curve?

   Answer:

$$AUC(C1) = 0.68$$

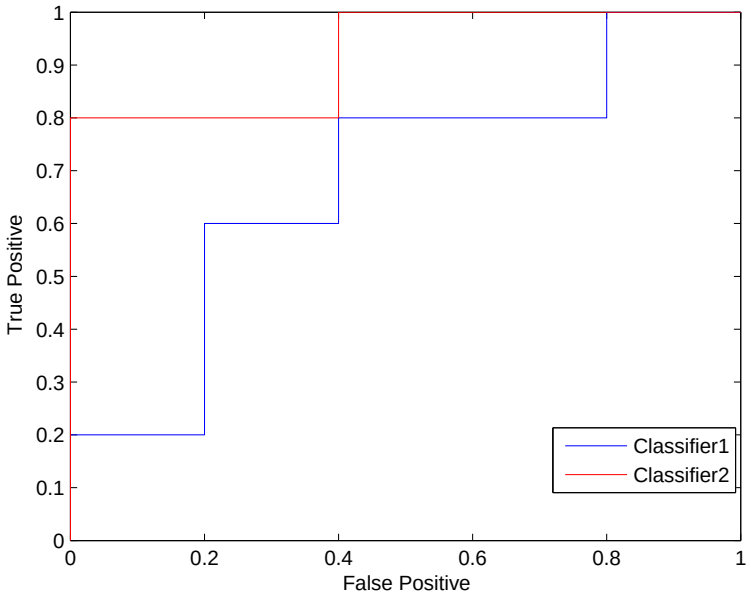


Figure 3.24. ROC curves for both classes.

$$AUC(C2) = 0.92$$

Hence, C2 has larger area under the ROC curve.

(c) Compute the Wilcoxon Mann Whitney statistic for both classifiers. The statistic can be computed as follows:

$$WMW = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x_i, y_j)}{mn}, \tag{3.10}$$

where

$$I(x, y) = \begin{cases} 1, & x_i > y_j; \\ 0, & \text{otherwise.} \end{cases}$$

Note that $\{x_0, x_1, \cdots, x_{m-1}\}$ correspond to the classifier outputs for the m positive examples while $\{y_0, y_1, \cdots, y_{n-1}\}$ correspond to the classifier outputs for the n negative examples (in this exercise, $m = n = 5$). Which classifier has a larger WMW value? Based on your answers, state the relationship between WMW and the ROC curve.

Answer:

$$WMW(C1) = \frac{1+3+4+4+5}{25} = 0.68$$

$$WMW(C2) = \frac{3+5+5+5+5}{25} = 0.92$$

Hence, C2 has larger WMW. WMW is equivalent to the area under ROC curve.

2. You have been asked to develop a classification model for diagnosing whether a patient is infected with a certain disease. To help you evaluate the models, your collaborator has provided you with a small test set ($N = 10$) with equal number of positive and negative examples. You applied two classifiers, C_1 and C_2. The outputs of the classifiers in terms of predicting whether each of the test examples belong to the "positive" class are summarized in the table below. The first row shows the probability a test example belongs to the positive class according to classifier C_1, while the second row shows the same information for classifier C_2. The last row indicates the true class label of the 10 test examples.

$P(y = +\|C_1)$	0.15	0.2	0.25	0.37	0.41	0.55	0.65	0.8	0.92	0.99
$P(y = +\|C_2)$	0.33	0.22	0.1	0.41	0.68	0.59	0.72	0.75	0.64	0.95
y	-	-	+	-	+	-	-	+	+	+

(a) Draw the corresponding ROC curves for both classifiers on the same plot. **Answer:** See Figure 3.25.

(b) For each classifier C_i, what is the optimal threshold we should use for $P(y = +|C_i)$ to obtain a high true positive rate and low false positive rate? Draw the confusion matrix associated with each classifier (using the selected optimal threshold for predicting positive class). Based on your results, which classifier is better in terms of (a) accuracy, and (b) F-measure?

Answer: For classifier C_1, the optimal threshold is when $P(y = +|C_1) \geq 0.7$. The confusion matrix for the classifier is

$P(y = +\|C_1) \geq 0.7$		Predicted	
		+	-
Actual	+	3	2
	-	0	5

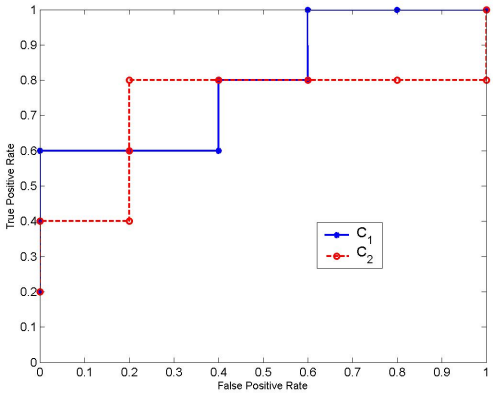


Figure 3.25. ROC curve for classifiers C_1 and C_2

F-measure $= \frac{2\times3}{2\times3+2+0} = \frac{6}{8} = 0.75$.
Accuracy $= \frac{8}{10} = 0.8$.
For classifier C_2, the optimal threshold is when $P(y = +|C_1) \geq 0.625$. The confusion matrix for the classifier is

$P(y = +\|C_2) \geq 0.625$		Predicted	
		+	-
Actual	+	4	1
	-	1	4

F-measure $= \frac{2\times4}{2\times4+1+1} = \frac{8}{10} = 0.8$.
Accuracy $= \frac{8}{10} = 0.8$.
C_2 is better than C_1 in terms of F-measure but they both have the same accuracy.

(c) Compute the area under ROC curve for each classifier. Which classifier has a larger area under the ROC curve?

Answer:

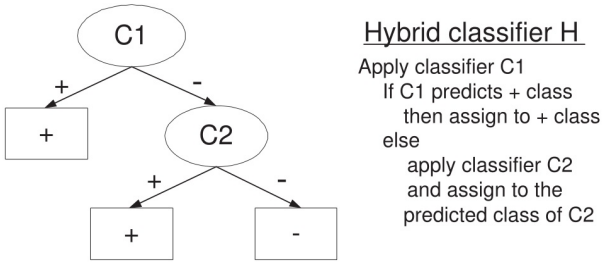
For C_1: Area under ROC curve $= 0.4\times0.6+0.2\times0.8+0.4\times1 = 0.8$.
For C_2: Area under ROC curve $= 0.2 \times 0.4 + 0.8 \times 0.8 = 0.72$.
Classifier C_1 has a larger area under ROC curve.

(d) Suppose a binary classifier produces a true positive rate of 40% and false positive rate of 60% (i.e., worse than random guessing).

Explain a simple approach you can use to improve the performance of the classifier so that it performs better than random guessing. What is the expected true positive rate and false positive rate of the classifier using your proposed approach?

Answer: A simple approach would be to predict the opposite of what the classifier says. For example, if the classifier predicts it to be a positive class, we should predict it as negative class instead. Similarly, if the classifier predicts it to be a negative class, we should declare it as positive. By reversing the prediction, the true positive rate of the classifier becomes 60% and its false positive rate becomes 40%.

3. Suppose we are given a pair of "independent" classifiers, $C_1$and C_2 (being independent means their errors are uncorrelated).



Assume the classifiers have been trained on a two-class problem (denoted as positive and negative class, respectively). The class distribution is skewed, i.e., the proportion of negative class outnumbers the positive class by 9:1. The precision and recall for classifier C_1 (with respect to the positive class) are 0.5 and 0.8, respectively. On the other hand, the precision and recall for classifier C_2 (with respect to the positive class) are both 0.6. Consider the hybrid classifier obtained by combining C_1 and C_2. Assume the precision and recall of C_2 remain unchanged when used in the hybrid setting (even though the class proportion has changed). Compare the F-measure of the hybrid classifier against that for C_1 and C_2. Which of them is the best? Show your steps clearly to receive full credit.

Answer: Suppose there are N total examples. Based on the information given, the confusion matrix for classifiers C_1 and C_2 are given below:

Classifier C_1		Predicted class +	Predicted class −	
Actual class	+	0.08N	0.02N	0.1N
	−	0.08N	0.82N	0.9N
		0.16N	0.84N	N

Precision = 0.5, Recall = 0.8, F-measure = $\frac{2rp}{r+p} = 0.6154$

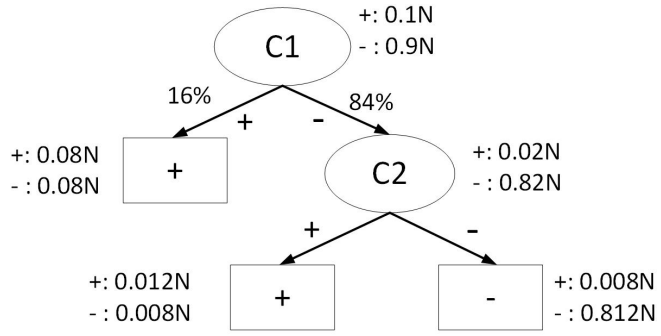
Classifier C_2		Predicted class +	Predicted class −	
Actual class	+	0.06N	0.04N	0.1N
	−	0.04N	0.86N	0.9N
		0.1N	0.9N	N

Precision = 0.6, Recall = 0.6, F-measure = $\frac{2rp}{r+p} = 0.6$

For the hybrid classifier, after applying C_1, 84% of the training examples will be propagated to C_2, out of which there are $0.02N$ positive and $0.82N$ negative examples. Based on this information, the confusion matrix for C_2 in the hybrid setting is given below (assuming it maintains the same precision and recall values as before):

Classifier C_2		Predicted class +	Predicted class −	
Actual class	+	0.012N	0.008N	0.02N
	−	0.008N	0.812N	0.82N
		0.02N	0.82N	0.84N

A summary of the class distribution at each leaf node in the hybrid classifier is shown in the Figure below.



The precision, recall, and F-measure for the hybrid classifier are:

Precision $= \frac{0.08N+0.012N}{0.08N+0.08N+0.012N+0.008N} = 0.5111$

Recall $= \frac{0.08N+0.012N}{0.1N} = 0.92$

F-measure $= \frac{2\times 0.5111\times 0.92}{0.5111+0.92} = 0.6571.$

This is better than the F-measure for both C_1 and C_2.

4. Consider the two decision trees shown in Figure 3.26 for a binary classification problem. Assume the classes are denoted as + and −, respectively. Suppose there are four binary attributes in the data, A, B, C, and D. The counts shown in the leaf nodes of the tree correspond to the number of training examples assigned to the nodes. Assume that the decision tree classifier assigns the majority class of training examples as the class label of each leaf node.

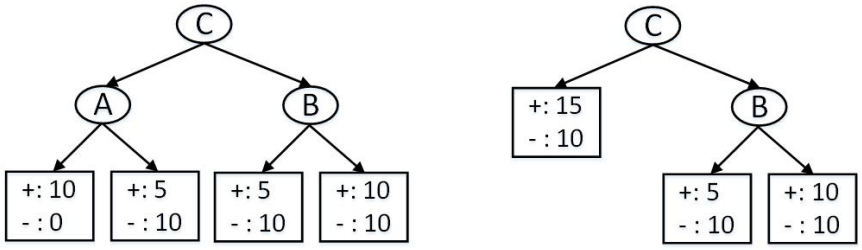


Figure 3.26. Two candidate decision trees

(a) Draw the confusion matrix for both trees on the training data. A confusion matrix is a table that summarizes the number of examples correctly or incorrectly predicted by the model. For example:

		Predicted	
		+	−
Actual	+	n_{++}	n_{+-}
	−	n_{-+}	n_{--}

In the above table, n_{+-} is the number of positive examples incorrectly predicted as negative class.

Answer: For the left tree, we may assign the class labels of the leaf nodes as follows (from left to right): +, -, -, +. In this case, the confusion matrix for the left tree is

C1		Predicted +	Predicted −
Actual	+	20	10
	−	10	20

If the right-most leaf node was assigned to the negative class, then the confusion matrix for the left tree is

C2		Predicted +	Predicted −
Actual	+	10	20
	−	0	30

For the right tree, we may assign the class labels of the leaf nodes as follows (from left to right): +, -, +. In this case, the confusion matrix for the left tree is

C3		Predicted +	Predicted −
Actual	+	25	5
	−	20	10

If the right-most leaf node was assigned to the negative class, then the confusion matrix for the right tree is

C4		Predicted +	Predicted −
Actual	+	15	15
	−	10	20

(b) Calculate the training error rate of both decision trees. Which tree has a lower training error?

Answer: The training error rate for the left tree is:

Training error $= \frac{20}{60} = 0.33$.

The training error rate for the right tree is:

Training error = $\frac{25}{60} = 0.42$.

Thus, the left tree has a lower error rate.

(c) Apply the minimum description length principle to determine which tree should be preferred.

Answer: To apply MDL, we first compute the following:

The cost for encoding each internal node = $\log_2 d = \log_2 4 = 2$ bits.

The cost for encoding each leaf node = $\log_2 c = \log_2 2 = 1$ bit.

The cost for encoding each error = $\log_2 N = \lceil \log_2 60 \rceil = 6$ bits.

Since the left tree has 3 internal nodes, 4 leaf nodes, and misclassifies 20 examples, its total description length is

$$3 \times 2 + 4 \times 1 + 20 \times 6 = 130 \text{ bits.}$$

Since the right tree has 2 internal nodes, 3 leaf nodes, and misclassifies 25 examples, its total description length is

$$2 \times 2 + 3 \times 1 + 25 \times 6 = 157 \text{ bits.}$$

Thus, according to the MDL principle, the left tree should be preferred.

4

Alternative Classification Techniques

4.1 Linear Classifier and SVM

1. Consider the 2-dimensional data shown in Figure 4.1. There are three data points, two of them are classified as positive (red circles) and one is negative (blue square). Let the decision boundary of the linear classifier be $w^T\mathbf{x} + b = 0$ (shown as a red line in the diagram).

   (a) The geometric margin of each data point is the perpendicular distance between each data point to the decision boundary (shown as d_1, d_2, and d_3, respectively). Derive an expression for the total geometric margin, $M = d_1 + d_2 + d_3$, as a function of w, c_1, c_2, and c_3.

   Answer:

   The three data points $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ satisfy the following equations:

$$\begin{aligned} w^T\mathbf{x}_1 + b &= -c_1 \\ w^T\mathbf{x}_2 + b &= c_2 \\ w^T\mathbf{x}_3 + b &= c_3 \end{aligned}$$

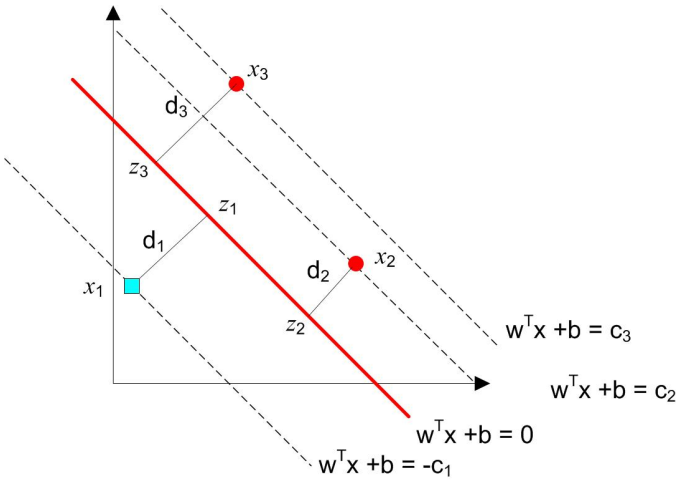


Figure 4.1. Linear classifier

Similarly, the corresponding three points on the decision boundary satisfy the following equations:

$$\begin{aligned} w^T \mathbf{z}_1 + b &= 0 \\ w^T \mathbf{z}_2 + b &= 0 \\ w^T \mathbf{z}_3 + b &= 0 \end{aligned}$$

Subtracting the equations, we obtain:

$$\begin{aligned} w^T(\mathbf{z}_1 - \mathbf{x}_1) &= c_1 \implies \|w\| d_1 = c_1 \\ w^T(\mathbf{x}_2 - \mathbf{z}_2) &= c_2 \implies \|w\| d_2 = c_2 \\ w^T(\mathbf{x}_3 - \mathbf{z}_3) &= c_3 \implies \|w\| d_3 = c_3 \end{aligned}$$

Putting them together, we obtain

$$M = \frac{c_1 + c_2 + c_3}{\|w\|}.$$

(b) If the decision boundary is shifted from $w^T\mathbf{x}+b = 0$ to $w^T\mathbf{x}+b' = 0$, where $b \neq b'$, how does it affect the total geometric margin, M?
Answer:

$$\begin{aligned} w^T(\mathbf{z}_1' - \mathbf{x}_1) + (b' - b) &= c_1 \implies \|w\|d_1 = c_1 + (b - b') \\ w^T(\mathbf{x}_2' - \mathbf{z}_2) + (b - b') &= c_2 \implies \|w\|d_2 = c_2 + (b' - b) \\ w^T(\mathbf{x}_3' - \mathbf{z}_3) + (b - b') &= c_3 \implies \|w\|d_3 = c_3 + (b' - b) \end{aligned}$$

The new total geometric margin is

$$M = \frac{c_1 + c_2 + c_3 + (b' - b)}{\|w\|}.$$

2. A linear classifier can be mathematically expressed as $f(\mathbf{x}) = \sum_i w_i \Phi_i(\mathbf{x})$, where each $\Phi_i(\mathbf{x})$ is a (possibly nonlinear) feature function of the original feature set $\mathbf{x}$. The predicted class for a test instance $\mathbf{x}$ is determined as follows:

$$\hat{y} = \begin{cases} +1, & \text{if } f(\mathbf{x}) \geq 0; \\ -1, & \text{otherwise.} \end{cases}$$

For each binary classification data set described below, state whether it can be perfectly classified by a linear classifier by choosing appropriate feature functions. If the answer is yes, write the mathematical expression for the linear classifier $f(\mathbf{x})$. Identify the feature functions $\Phi_i(\mathbf{x})$ as well as the parameters $\mathbf{w}$ in your expression.

(a) A data set with 4 continuous-valued features x_1, x_2, x_3, and x_4. The class label is +1 if the product of the x_1 and x_2 is greater than or equal to the product of x_3 and x_4; otherwise, it is -1.

Answer: Yes. The data can be perfectly classified as follows

$$f(x) = x_1x_2 - x_3x_4 = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1x_2 \\ x_3x_4 \end{pmatrix}$$

By choosing w=[1 -1], $\Phi_1(\mathbf{x}) = x_1x_2$ and $\Phi_2(\mathbf{x}) = x_3x_4$, we can construct a linear classifier using the two-dimensional feature function $\Phi = [\Phi_1\ \Phi_2]$.

(b) A data set with 4 continuous-valued features x_1, x_2, x_3, and x_4. The class label is +1 if at least one of the features is greater than 10; otherwise, it is -1.

Answer: Yes. The data can be perfectly classified as follows

$$f(x) = sign(x_1-10)+sign(x_2-10)+sign(x_3-10)+sign(x_4-10)-0.5,$$

where $sign(z) = 1$ if $z > 0$ and 0 otherwise. By choosing w=[-0.5 1 1 1 1] and $\Phi_0 = 1, \Phi_i(\mathbf{x}) = sign(x_i - 10)$, we can construct a linear classifier to perfectly classify the data.

(c) A data set with 2 binary features, x_1 and x_2, whose class label y is determined as follows (this is similar to the exclusive OR binary operator using -1 instead of 0):

x_1	x_2	y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Answer: Yes. The data can be perfectly classified as follows

$$f(x) = -x_1x_2$$

By choosing w=-1 and the feature function $\Phi(x) = x_1x_2$, we can construct a linear classifier to perfectly classify the data.

3. A linear classifier can be mathematically expressed as $f(\mathbf{x}) = \sum_i w_i\Phi_i(\mathbf{x})$, where each $\Phi_i(\mathbf{x})$ is a (possibly nonlinear) feature function of the original feature set $\mathbf{x}$. The predicted class for a test instance $\mathbf{x}$ is determined as follows:

$$\hat{y} = \begin{cases} +1, & \text{if } f(\mathbf{x}) \geq 0; \\ -1, & \text{otherwise.} \end{cases}$$

For each binary classification data set described below, state whether it can be perfectly classified by a linear classifier by choosing appropriate feature functions. If the answer is yes, write the mathematical expression for the linear classifier $f(\mathbf{x})$. Identify the feature functions $\Phi_i(\mathbf{x})$ as well as the parameters $\mathbf{w}$ in your expression.

(a) A data set with 4 continuous-valued features x_1, x_2, x_3, and x_4. The class label is +1 if the average of the first two features is greater than or equal to the average of the last two features; otherwise, it is −1.

Answer: Yes. We can choose $\Phi_i(\mathbf{x}) = x_i$, i.e., use the original features as feature function. The linear classifier can be written as

$$f(\mathbf{x}) = \frac{x_1 + x_2 - x_3 - x_4}{2}$$

(b) A data set with 8 Boolean features, where each x_i can take the value 0 or 1. The class label is +1 if there are more 1s than 0s; otherwise it is −1.

Answer: Yes. We can choose $\Phi_i(\mathbf{x}) = x_i$ $(\forall i \in \{1, 8\})$ and $\Phi_9(\mathbf{x}) = 1$. The linear classifier can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{8} x_i - 4.5$$

The weights are equal to 1 for the first 8 feature functions and -4.5 for the last one.

(c) A data set with 8 Boolean features, where each x_i can take the value 0 or 1. The class label is +1 if there are even number of 1s; otherwise it is −1. (0 is considered an even number as well).

Answer: Yes. First, convert the Boolean features into $\{-1, +1\}$ by setting $2x_i - 1$. If there are even number of 1s, then the product $2x_i - 1$ of all the features should be non-negative. Thus, we can define the linear classifier as follows:

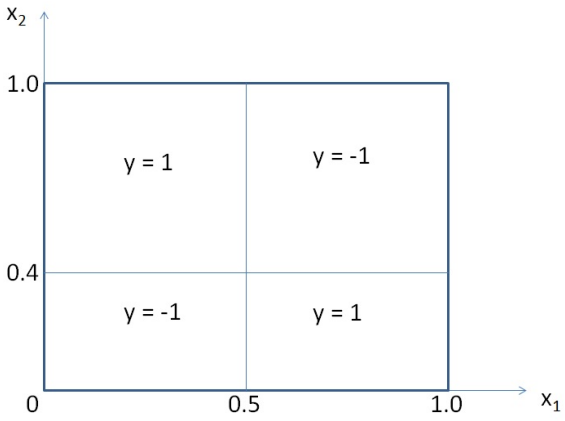
$$f(\mathbf{x}) = \prod_{i=1}^{8}(2x_i - 1) = 256\Phi_0(\mathbf{x}) - 128\Phi_1(\mathbf{x}) - 128\Phi_2(\mathbf{x}) + \cdots + 1$$

We can define the following feature functions, $\Phi_i(\mathbf{x}) = x_1^{I_1} x_2^{I_2} \cdots x_8^{I_8}$, where each $I_j \in \{0, 1\}$. For example, $\Phi_0(\mathbf{x}) = x_1x_2x_3x_4x_5x_6x_7x_8$, $\Phi_1(\mathbf{x}) = x_1x_2x_3x_4x_5x_6x_7$, and $\Phi_2(\mathbf{x}) = x_1x_2x_3x_4x_5x_6x_8$. The weights correspond to the polynomial expansion of the product of $(2x_i - 1)$.

(d) A data set with 2 continuous-valued features x_1 and x_2, where the range of possible values for each x_i is between 0 and 1. The class labels are shown in the diagram below.

Answer: Yes, the linear classifier is given by

$$f(\mathbf{x}) = -(x_1 - 0.5)(x_2 - 0.4) = -x_1x_2 + 0.4x_1 + 0.5x_2 - 0.2,$$



where we choose $\Phi_1(\mathbf{x}) = x_1x_2$, $\Phi_2(\mathbf{x}) = x_1$, $\Phi_3(\mathbf{x}) = x_2$, and $\Phi_0(\mathbf{x}) = 1$ and $w_1 = -1$, $w_2 = 0.4$, $w_3 = 0.5$, and $w_0 = -0.2$. This is because for the data points that belong to the class $y = -1$, then $(x_1 - 0.5)$ and $(x_2 - 0.4)$ must be both positive or both negative.

4. Consider the following loss function for a binary classification problem:

$$E(\mathbf{w}) = \sum_i (1 - y_i \mathbf{w}^T \mathbf{x}_i) \text{ s.t. } \|\mathbf{w}\|_2^2 = 1,$$

where $y_i \in \{-1, +1\}$. This is quite similar to the perceptron loss function, except the sum is taken over all the training examples (instead of summing over the misclassified training examples only).

Derive a closed-form solution for $\mathbf{w}$ that minimizes the constrained optimization problem. Hint: see lecture 5 on how to solve a constrained optimization problem with the Lagrange multiplier method. Use the constraint $\|\mathbf{w}\|_2^2 = 1$ to eliminate the Lagrange multiplier from your solution for $\mathbf{w}$.

Answer: First, the Lagrangian for the problem is

$$\mathcal{L} = \sum_i (1 - y_i \mathbf{w}^T \mathbf{x}_i) + \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

After taking its partial derivative with respect to $\mathbf{w}$, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = -\sum_i y_i \mathbf{x}_i + 2\lambda \mathbf{w} &= 0 \\ \implies \mathbf{w} &= \frac{1}{2\lambda}\mathbf{X}^T\mathbf{y} \end{aligned} \tag{4.1}$$

where we have used a matrix notation on the second line. Furthermore

$$\mathbf{w}^T\mathbf{w} = \frac{1}{4\lambda^2}\mathbf{y}^T\mathbf{X}\mathbf{X}^T\mathbf{y} = 1,$$

where we have applied the equality constraint $\mathbf{w}^T\mathbf{w} = 1$. Thus, the Lagrange parameter is found to be

$$\lambda = \frac{1}{2}\sqrt{\mathbf{y}^T\mathbf{X}\mathbf{X}^T\mathbf{y}}$$

Replacing this back into (4.1), we have

$$\mathbf{w} = \frac{\mathbf{X}^T\mathbf{y}}{\sqrt{\mathbf{y}^T\mathbf{X}\mathbf{X}^T\mathbf{y}}}$$

5. A linear classifier in SVM can be mathematically expressed as $f(\mathbf{x}) = \sum_i w_i \Phi_i(\mathbf{x})$, where each $\Phi_i(\mathbf{x})$ is a feature function of the original feature set $\mathbf{x}$. Note that the transformed feature set $\{\Phi_i(\mathbf{x})\}$ could be infinite-dimensional. The predicted class for a test instance $\mathbf{x}$ is determined as follows:

$$\hat{y} = \begin{cases} +1, & \text{if } f(\mathbf{x}) \geq 0; \\ -1, & \text{otherwise.} \end{cases}$$

For each binary classification data set described below, state whether it can be perfectly classified by a linear classifier by choosing appropriate feature functions. Restrict the feature functions to polynomial expansions of the original attributes, e.g., $\Phi_1(\mathbf{x}) = x_1 x_2^2$ or $\Phi_2(\mathbf{x}) = x_2 x_3 x_4$, where x_1, x_2, x_3, and x_4 are part of the original attributes. If the answer is yes, write the mathematical expression for the linear classifier $f(\mathbf{x})$. Identify the feature functions $\Phi_i(\mathbf{x})$ as well as the parameters $\mathbf{w}$ in your expression.

(a) A data set with 4 continuous-valued features x_1, x_2, x_3, and x_4. The class label is +1 if the product of the x_1 and x_2 is greater than or equal to the product of x_3 and x_4; otherwise, it is −1.

Answer: Yes, the data can be perfectly classified by choosing the following feature functions: $\Phi_1(\mathbf{x}) = x_1x_2$ and $\Phi_2(\mathbf{x}) = x_3x_4$.

$$f(\mathbf{x}) = w_1\Phi_1(\mathbf{x}) + w_2\Phi_2(\mathbf{x}).$$

The weights of the feature functions are $w_1 = 1$ and $w_2 = -1$.

(b) A data set with 4 continuous-valued features x_1, x_2, x_3, and x_4. The class label is +1 if at least one of the features is greater than 10; otherwise, it is −1.

Answer:

Let $\mathbf{I}(\cdot)$ be an indicator function, whose value is defined as follows:

$$I(x) = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

The classification problem can be solved by defining

$$f(\mathbf{x}) = I(x_1 - 10) + I(x_2 - 10) + I(x_3 - 10) + I(x_4 - 10) - 0.5,$$

where $\mathbf{x} = (x_1, x_2, x_3, x_4)$. However, our objective is to express $f(\mathbf{x})$ as a linear combination of polynomial feature functions. To do this, we can approximate the indicator functions by sigmoid function:

$$I(x) \to \sigma(x) = \frac{1}{1 + e^{-\alpha x}}$$

Figure 4.2 illustrates an example of the sigmoid function for different values of α. As α goes to ∞, the sigmoid function approaches the indicator function.

Furthermore, the sigmoid function can be expressed as a polynomial expansion using Maclaurin series, which is a special case of Taylor series centered at zero (see, for example, `http://mathworld.wolfram.com/SigmoidFunction.html`):

$$\sigma(x) = \frac{1}{1 + e^{-\alpha x}} = \sum_{n=0}^{\infty} \frac{(-1)^n E_n(0)}{2n!}(\alpha x)^n = 0.5 + \sum_{n=1}^{\infty} J(n)x^n,$$

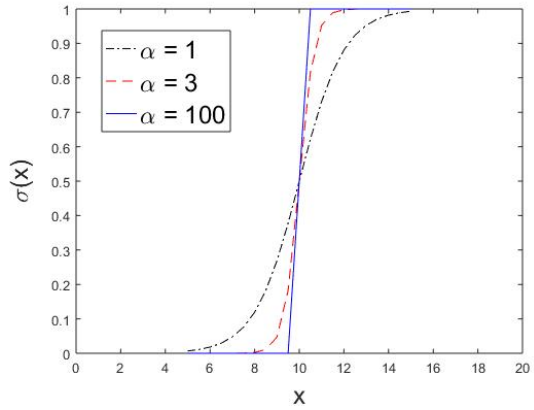


Figure 4.2. Sigmoid function

where $E_n(x)$ is an Euler polynomial. Thus, in principle, given a finite training set, we could find an infinite number of feature functions that perfectly classify the data by choosing the appropriate sigmoid function and its Maclaurin series to mimic the indicator functions. The feature functions are simply

$$\{\Phi_i\} = \{1, x_1, x_2, x_3, x_4, x_1^2, x_2^2, x_3^2, x_4^2, \cdots\}$$

and their corresponding weights are

$$(1.5, J(1), J(1), J(1), J(1), J(2), J(2), J(2), J(2), \cdots)$$

(c) A data set with 2 continuous-valued features x_1 and x_2. The class label is $+1$ if the exponential value of the difference between $x_1 - x_2$ is greater than 100 (i.e., $\exp[x_1 - x_2]$); otherwise, it is -1.
Answer: The classification problem can be solved by defining

$$f(\mathbf{x}) = \exp[x_1 - x_2] - 100$$

Since the Taylor series expansion for $\exp(\cdot)$ is

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!},$$

we can define the following polynomial feature functions:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{n=0}^{\infty} \frac{(x_1 - x_2)^n}{n!} - 100 \\ &= \sum_{n=1}^{\infty} J_1(n) x_1^n + \sum_{n=1}^{\infty} J_2(n) x_2^n + \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} J_3(n,m) x_1^n x_2^m - 99 \end{aligned}$$

Thus, we can perfectly classify the data by choosing the following feature functions

$$\{1, x_1, x_2, x_1 x_2, x_1 x_2^2, x_1^2 x_2^2, \cdots\}$$

with their corresponding weights given by

$$(-99, J_1(1), J_2(1), J_3(1,1), J_3(1,2), J_3(2,1), \cdots)$$

(d) A data set with 2 binary features, x_1 and x_2, whose class label y is determined as follows (this is similar to the exclusive OR binary operator using -1 instead of 0):

x_1	x_2	y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Answer:
Yes. The data can be perfectly classified as follows

$$f(\mathbf{x}) = -x_1 x_2$$

By choosing $w = -1$ and the feature function $\Phi(\mathbf{x}) = x_1 x_2$, we can construct a linear classifier to perfectly classify the data.

4.2 Bayesian classifiers

1. Consider a training set with 3 features, X_1, X_2, and X_3, for a binary classification problem. The class distribution is shown in the table below.

X_1	X_2	X_3	Number of positive examples	Number of negative examples
1	1	1	20	8
1	0	0	20	17
0	1	0	5	8
0	0	0	5	17

(a) Based on the information above, determine whether X_1 and X_2 are independent of each other.

Answer: If X_1 and X_2 are independent, then $P(X_1, X_2) = P(X_1)P(X_2)$ for ALL possible values of X_1 and X_2. From the training data:

$$p(X_1 = 1, X_2 = 1) \quad = \quad \frac{20 + 8}{100} = 0.28$$

whereas

$$p(X_1 = 1)p(X_2 = 1) = \frac{28 + 37}{100} \times \frac{28 + 13}{100} = 0.2665,$$

Since $p(X_1 = 1)p(X_2 = 1) \neq p(X_1 = 1, X_2 = 1)$, X_1 and X_2 are not independent of each other. Note: you only need to find a counter-example to disprove the independence relationship.

(b) Determine whether X_1 and X_2 are conditionally independent of each other given the class.

Answer:

We can estimate the class conditional probabilities from the training data to obtain the following:

$$\begin{aligned}
&p(X_1 = 1|+) = 0.8,\ p(X_2 = 1|+) = 0.5,\\
&p(X_1 = 0|+) = 0.2,\ p(X_2 = 0|+) = 0.5\\
&p(X_1 = 1, X_2 = 1|+) = 0.4,\ p(X_1 = 1, X_2 = 0|+) = 0.4,\\
&p(X_1 = 0, X_2 = 1|+) = 0.1,\ p(X_1 = 0, X_2 = 0|+) = 0.1\\
&p(X_1 = 1|-) = 0.5,\ p(X_2 = 1|-) = 0.32,\\
&p(X_1 = 0|-) = 0.5,\ p(X_2 = 0|-) = 0.68\\
&p(X_1 = 1, X_2 = 1|-) = 0.16, p(X_1 = 1, X_2 = 0|-) = 0.34,\\
&p(X_1 = 0, X_2 = 1|+) = 0.16, p(X_1 = 0, X_2 = 0|+) = 0.34.
\end{aligned}$$

Since $p(X_1, X_2|+) = p(X_1|+)p(X_2|+)$ and $p(X_1, X_2|-) = p(X_1|+)p(X_2|-)$ for all X_1 and X_2, X_1 and X_2 are therefore conditionally independent given the class. Note: it is insufficient to show it only for $X_1 = 1$ and $X_2 = 1$ but not other values.

(c) Compute the class conditional probabilities $P(X_1 = 1|+)$, $P(X_1 = 1|-)$, $P(X_2 = 1|+)$, $P(X_2 = 1|-)$, $P(X_3 = 1|+)$, and $P(X_3 = 1|-)$.

Answer:

$P(X_1 = 1|+) = 0.8$, $P(X_1 = 1|-) = 0.5$, $P(X_2 = 1|+) = 0.5$, $P(X_2 = 1|-) = 0.32$, $P(X_3 = 1|+) = 0.4$, and $P(X_3 = 1|-) = 0.16$

(d) Use the class conditional probabilities given in the previous question to predict the class label of each example with the feature set given in the training set above. Use your results to compute the training error rate of the naïve Bayes classifier.

Answer: The naïve Bayes classifier makes a prediction based on the following equation:

$$\begin{aligned} \hat{y} &= \arg\max_y p(y|X_1, X_2, X_3) \\ &= \arg\max_y p(X_1|y)p(X_2|y)p(X_3|y)p(y) \end{aligned}$$

You need to determine the predicted class for all 4 cases of the training examples:

i. For $X_1 = X_2 = X_3 = 1$,

$$p(X_1 = 1|+)p(X_2 = 1|+)p(X_3 = 1|+)p(+) = 0.8{\times}0.5{\times}0.4{\times}0.5 = 0.08$$

$$p(X_1 = 1|-)p(X_2 = 1|-)p(X_3 = 1|-)p(-) = 0.5{\times}0.32{\times}0.16{\times}0.5 = 0.0128$$

Hence, $X_1 = X_2 = X_3 = 1$ will be labeled as $+$.

ii. For $X_1 = 1$, $X_2 = X_3 = 0$,

$$p(X_1 = 1|+)p(X_2 = 0|+)p(X_3 = 0|+)p(+) = 0.8{\times}0.5{\times}0.6{\times}0.5 = 0.12$$

$$p(X_1 = 1|-)p(X_2 = 0|-)p(X_3 = 0|-)p(-) = 0.5{\times}0.68{\times}0.84{\times}0.5 = 0.1428$$

Hence, $X_1 = 1$, $X_2 = X_3 = 0$ will be labeled as $-$.

iii. For $X_1 = 0$, $X_2 = 1$, $X_3 = 0$,

$$p(X_1 = 0|+)p(X_2 = 1|+)p(X_3 = 0|+)p(+) = 0.2{\times}0.5{\times}0.6{\times}0.5 = 0.03$$

$$p(X_1 = 0|-)p(X_2 = 1|-)p(X_3 = 0|-)p(-) = 0.5{\times}0.32{\times}0.84{\times}0.5 = 0.0672$$

Hence, $X_1 = 0$, $X_2 = 1$, $X_3 = 0$ will be labeled as $-$.

iv. For $X_1 = X_2 = X_3 = 0$,

$$p(X_1 = 0|+)p(X_2 = 0|+)p(X_3 = 0|+)p(+) = 0.2\times0.5\times0.6\times0.5 = 0.03$$

$$p(X_1 = 0|-)p(X_2 = 0|-)p(X_3 = 0|-)p(-) = 0.5\times0.68\times0.84\times0.5 = 0.1428$$

Hence, $X_1 = X_2 = X_3 = 0$ will be labeled as $-$.

If you apply the classifier to the training examples given, 38 out of 100 examples will be misclassified. So the training error rate is 0.38.

2. Consider the directed acyclic graph shown in Figure 4.3. Determine whether each of the following independence or conditional independence assumptions are valid according to the constraints given by the graph. To receive full credit, make sure you show your steps clearly (to prove/disprove the assumptions).

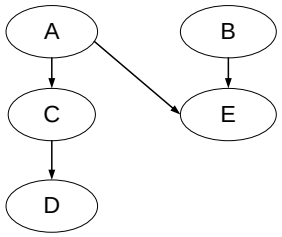


Figure 4.3. A directed acyclic graph.

(a) D $\perp$ E

Answer: First you need to establish the relationship between $P(D, E)$ and $P(D)P(E)$ in order to determine whether they are independent.

$$\begin{aligned} P(D,E) &= \sum_{A,B,C} P(A,B,C,D,E) \\ &= \sum_{A,B,C} P(D|C)P(C|A)P(E|A,B)P(A)P(B) \end{aligned} \quad (4.3)$$

Furthermore, we can show that

$$\begin{aligned} P(D) &= \sum_{A,B,C,E} P(A,B,C,D,E) \\ &= \sum_{A,B,C,E} P(D|C)P(C|A)P(E|A,B)P(A)P(B) \\ &= \sum_{A,B,C} P(D|C)P(C|A)P(A)P(B) \\ &= \sum_{A,C} P(D|C)P(C|A)P(A) \\ &= \sum_{C} P(D|C)P(C) \text{ where } \sum_{A} P(C|A)P(A) = P(C) \end{aligned}$$

and

$$\begin{aligned} P(E) &= \sum_{A,B,C,D} P(A,B,C,D,E) \\ &= \sum_{A,B,C,D} P(D|C)P(C|A)P(E|A,B)P(A)P(B) \\ &= \sum_{A,B,C} P(C|A)P(E|A,B)P(A)P(B) \\ &= \sum_{A,B} P(E|A,B)P(A)P(B) \end{aligned}$$

Therefore

$$\begin{aligned} P(D)P(E) &= \Big(\sum_{C} P(D|C)P(C)\Big)\Big(\sum_{A,B} P(E|A,B)P(A)P(B)\Big) \\ &= \sum_{ABC} P(D|C)P(C)P(E|A,B)P(A)P(B) \qquad (4.4) \end{aligned}$$

If we compare against $P(D,E)$ given in Equation (4.3), $P(C|A)$ must be equal to $P(C)$ in order for D and E to be independent. Clearly this contradicts with the directed acyclic graph since there is a direct link from node A to C. Thus, $P(D,E) \neq P(D)P(E)$, which means D and E are not independent of each other.

(b) C $\perp$ E | A, B

Answer:

$$\begin{aligned}
P(C,E|A,B) &= \frac{P(A,B,C,E)}{P(A,B)} \\
&= \frac{\sum_D P(A,B,C,D,E)}{\sum_{C,D,E} P(A,B,C,D,E)} \\
&= \frac{\sum_D P(D|C)P(C|A)P(E|A,B)P(A)P(B)}{\sum_{C,D,E} P(D|C)P(C|A)P(E|A,B)P(A)P(B)} \\
&= \frac{P(C|A)P(E|A,B)P(A)P(B)}{\sum_{C,E} P(C|A)P(E|A,B)P(A)P(B)} \\
&= \frac{P(C|A)P(E|A,B)P(A)P(B)}{P(A)P(B)} \\
&= P(C|A)P(E|A,B)
\end{aligned}$$

Furthermore it can be shown that

$$\begin{aligned}
P(C|A,B) &= \frac{P(A,B,C)}{P(A,B)} \\
&= \frac{\sum_{D,E} P(A,B,C,D,E)}{\sum_{C,D,E} P(A,B,C,D,E)} \\
&= \frac{\sum_{D,E} P(D|C)P(C|A)P(E|A,B)P(A)P(B)}{\sum_{C,D,E} P(D|C)P(C|A)P(E|A,B)P(A)P(B)} \\
&= \frac{\sum_E P(C|A)P(E|A,B)P(A)P(B)}{\sum_{C,E} P(C|A)P(E|A,B)P(A)P(B)} \\
&= \frac{P(C|A)P(A)P(B)}{P(A)P(B)} \\
&= P(C|A)
\end{aligned}$$

Since $P(C,E|A,B) = P(C|A,B)P(E|A,B)$, C and E are conditionally independent given A and B.

(c) C $\perp$ E | D

Answer:

$$
\begin{aligned}
P(C,E|D) &= \frac{P(C,D,E)}{P(D)} \\
&= \frac{\sum_{A,B} P(A,B,C,D,E)}{P(D)} \\
&= \frac{\sum_{A,B} P(D|C)P(C|A)P(E|A,B)P(A)P(B)}{P(D)} \\
&= \frac{P(D|C)\sum_{A,B} P(C|A)P(E|A,B)P(A)P(B)}{P(D)} \\
&= \frac{P(C|D)\sum_{A,B} P(C|A)P(E|A,B)P(A)P(B)}{P(C)} \\
&= P(C|D)\left(\frac{\sum_{A,B} P(C|A)P(E|A,B)P(A)P(B)}{P(C)}\right) \\
&= P(C|D)\left(\frac{\sum_{A,B} P(A|C)P(E|A,B)P(A)P(B)}{P(A)}\right) \\
&= P(C|D)\left(\sum_{A,B} P(A|C)P(E|A,B)P(B)\right) \qquad (4.5)
\end{aligned}
$$

where we have used the following equations (from Bayes theorem)

$$\frac{P(D|C)}{P(D)} = \frac{P(C|D)}{P(C)}, \qquad \frac{P(C|A)}{P(C)} = \frac{P(A|C)}{P(A)}.$$

If C and E are conditionally independent given D, then the term in parenthesis on the right-hand side of the equation must be equal to $P(E|D)$. However,

$$
\begin{aligned}
P(E|D) &= \frac{P(D,E)}{P(D)} \\
&= \frac{\sum_{A,B,C} P(A,B,C,D,E)}{P(D)} \\
&= \frac{\sum_{A,B,C} P(D|C)P(C|A)P(E|A,B)P(A)P(B)}{P(D)} \\
&= \frac{\sum_{A,B,C} P(C|D)P(A|C)P(E|A,B)P(A)P(B)}{P(A)} \\
&= \sum_{C} P(C|D)\left(\sum_{A,B} P(A|C)P(E|A,B)P(B)\right) \qquad (4.6)
\end{aligned}
$$

Comparing Equation (4.6) against Equation (4.5), it is clear that if the parenthesis term on the right side of Equation (4.5) is equal to $P(E|D)$ then

$$\sum_{A,B} P(A|C)P(E|A,B)P(B) = \sum_{A,B,C} P(C|D)P(A|C)P(E|A,B)P(B),$$

which is not always true. Therefore, since we cannot conclude that $P(C,E|D) = P(C|D)P(E|D)$, C and E are not conditionally independent given D.

3. Consider the following training set, which contains 3 binary attributes X_1, X_2, and X_3. There are 50 examples in the training set, with equal number of positive and negative examples.

X_1	X_2	X_3	Number of positive training examples	Number of negative training examples
1	1	1	5	0
1	0	1	10	10
0	0	1	5	5
0	1	1	0	10
0	0	0	5	0

(a) Compute the class conditional probabilities $P(X_1 = 1|+)$, $P(X_1 = 1|-)$, $P(X_2 = 1|+)$, $P(X_2 = 1|-)$, $P(X_3 = 1|+)$, and $P(X_3 = 1|-)$.
Solution:

$$\begin{aligned} P(X_1 = 1|+) &= (5+10)/25 = 0.6 \\ P(X_1 = 1|-) &= (0+10)/25 = 0.4 \\ P(X_2 = 1|+) &= (5+0)/25 = 0.2 \\ P(X_2 = 1|-) &= (0+10)/25 = 0.4 \\ P(X_3 = 1|+) &= (5+10+5+0)/25 = 0.8 \\ P(X_3 = 1|-) &= (10+5+10)/25 = 1 \end{aligned}$$

(b) Use the class conditional probabilities given in the previous question to predict the class label of each example with the feature set given in the training set above. Use your results to compute the training error rate of the naïve Bayes classifier.

Solution:
We need to compare $P(+|X)$ and $P(-|X)$, and the class with bigger conditional probability is the predicted class.

$$P(+|X) = P(+|X_1, X_2, X_3) = \frac{P(X_1, X_2, X_3|+) \times P(+)}{P(X_1, X_2, X_3)}$$
$$P(-|X) = P(-|X_1, X_2, X_3) = \frac{P(X_1, X_2, X_3|-) \times P(-)}{P(X_1, X_2, X_3)}$$

and,

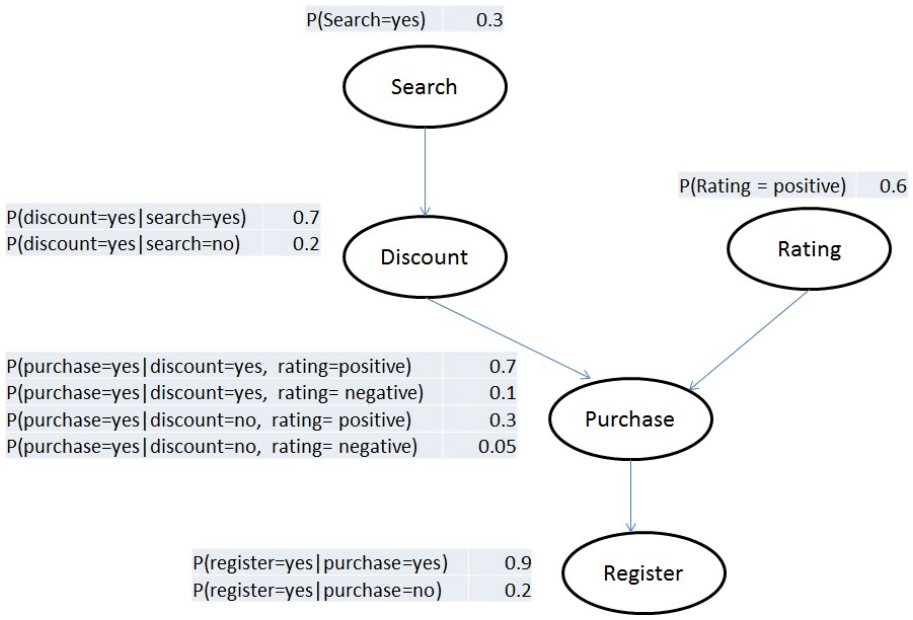
$$P(+) = P(-) = 25/50 = 0.5$$

The problem can be simplified to compare $P(X_1, X_2, X_3|+)$ against $P(X_1, X_2, X_3|-)$.

$$\begin{aligned}
P(X_1 = 1, X_2 = 1, X_3 = 1|+) &= 0.6 \times 0.2 \times 0.8 = 0.096 \\
P(X_1 = 1, X_2 = 1, X_3 = 1|-) &= 0.4 \times 0.4 \times 1 = 0.16 \\
P(X_1 = 1, X_2 = 0, X_3 = 1|+) &= 0.6 \times 0.8 \times 0.8 = 0.384 \\
P(X_1 = 1, X_2 = 0, X_3 = 1|-) &= 0.4 \times 0.6 \times 1 = 0.24 \\
P(X_1 = 0, X_2 = 0, X_3 = 1|+) &= 0.4 \times 0.8 \times 0.8 = 0.256 \\
P(X_1 = 0, X_2 = 0, X_3 = 1|-) &= 0.6 \times 0.6 \times 1 = 0.36 \\
P(X_1 = 0, X_2 = 1, X_3 = 1|+) &= 0.4 \times 0.2 \times 0.8 = 0.064 \\
P(X_1 = 0, X_2 = 1, X_3 = 1|-) &= 0.6 \times 0.4 \times 1 = 0.24 \\
P(X_1 = 0, X_2 = 0, X_3 = 0|+) &= 0.4 \times 0.8 \times 0.2 = 0.064 \\
P(X_1 = 0, X_2 = 0, X_3 = 0|-) &= 0.6 \times 0.6 \times 0 = 0
\end{aligned}$$

Thus,
$X = (1, 1, 1)$ is predicted as "-". 5 errors are made.
$X = (1, 0, 1)$ is predicted as "+". 10 errors are made.
$X = (0, 0, 1)$ is predicted as "-". 5 errors are made.
$X = (0, 1, 1)$ is predicted as "-". 0 errors are made.
$X = (0, 0, 0)$ is predicted as "+". 0 errors are made.
So, error rate is $(5 + 10 + 5)/50 = 0.4$.

4. Consider the problem of predicting whether an online visitor will purchase a product on a website. A visitor may search for the product they



want to buy or reach the product page through casual browsing starting from the homepage. Some product pages have discounts to entice the visitor to make a purchase while others do not. Visitors may also look at the overall rating of the product before deciding whether to buy the product. Visitors who are ready to make a purchase are also given a choice whether to register as a member or remain as a guest to the website. The above figure shows a Bayesian network that captures the relationships among the various attributes. Use the Bayesian network to answer the following questions.

(a) Determine whether each of the following independence or conditional independence assumptions are valid according to the constraints given by the graph shown in Figure **??**.

i. Search $\perp$ Rating
Answer: Let S: search, R: rating, D: discount, P: purchase, and G: register.

$$
\begin{aligned}
P(S,R) &= \sum_{DPG} P(S,R,D,P,G) \\
&= \sum_{DPG} P(S)P(R)P(D|S)P(P|D,R)P(G|P) \quad \text{(from the dag)} \\
&= P(S)P(R)\sum_{D} P(D|S)\sum_{P} P(P|D,R)\sum_{G} P(G|P) \\
&= P(S)P(R)\sum_{D} P(D|S)\sum_{P} P(P|D,R) \\
&= P(S)P(R)\sum_{D} P(D|S) \\
&= P(S)P(R), \qquad (4.7)
\end{aligned}
$$

where $\sum_G P(G|P) = \sum_P P(P|D,R) = \sum_D P(D|S) = 1$. Thus Search is independent of Rating.

ii. Register $\perp$ Search | Discount, Rating
Answer:

$$
\begin{aligned}
P(S,G|D,R) &= \frac{P(S,G,D,R)}{P(D,R)} \\
&= \frac{\sum_P P(S,R,D,P,G)}{P(D,R)} \\
&= \frac{\sum_P P(S)P(R)P(D|S)P(P|R,D)P(G|P)}{P(D,R)} \\
&= \frac{P(S)P(R)P(D|S)\sum_P P(P|R,D)P(G|P)}{P(D,R)} \\
&= \frac{P(S)P(R)P(D|S)}{P(D,R)} \times \sum_P P(P|R,D)P(G|P) \qquad (4.8)
\end{aligned}
$$

Furthermore,

$$\begin{aligned} P(S|D,R) &= \frac{P(S,D,R)}{P(D,R)} \\ &= \frac{\sum_{P,G} P(S,R,D,P,G)}{P(D,R)} \\ &= \frac{\sum_{P,G} P(S)P(R)P(D|S)P(P|R,D)P(G|P)}{P(D,R)} \\ &= \frac{\sum_{P} P(S)P(R)P(D|S)P(P|R,D)}{P(D,R)} \\ &= \frac{P(S)P(R)P(D|S)}{P(D,R)} \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} P(G|D,R) &= \frac{P(G,D,R)}{P(D,R)} \\ &= \frac{\sum_{P,S} P(S,R,D,P,G)}{P(D,R)} \\ &= \frac{\sum_{P,S} P(S)P(R)P(D|S)P(P|R,D)P(G|P)}{P(D,R)} \\ &= \frac{\sum_{S} P(S)P(R)P(D|S)}{P(D,R)} \times \sum_{P} P(P|R,D)P(G|P) \\ &= \sum_{S} P(S|D,R) \times \sum_{P} P(P|R,D)P(G|P) \quad \text{(based on Equation (4.9))} \\ &= 1 \times \sum_{P} P(P|R,D)P(G|P) \\ &= \sum_{P} P(P|R,D)P(G|P) \end{aligned} \tag{4.10}$$

Plugging in Equations (4.9) and (4.10) into (4.8), we obtain

$$P(S,G|D,R) = P(S|D,R) \times P(G|D,R)$$

Therefore Register is conditionally independent of Search given Rating and Discount.

(b) Use the Bayesian network to predict whether a visitor who search for a product with an overall positive rating is likely to buy the product at the web site.

Answer: For this question, you need to compare $P(P = yes|R = +)$ against $P(P = no|R = +)$.

$$\begin{aligned}
\frac{P(P=yes|R=+)}{P(P=no|R=+)} &= \frac{P(P=yes,R=+)}{P(P=no,R=+)} \\
&= \frac{\sum_{S,G,D} P(P=yes,R=+,S,G,D)}{\sum_{S,G,D} P(P=no,R=+,S,G,D)} \\
&= \frac{\sum_{S,G,D} P(S)P(R=+)P(D|S)P(P=yes|R=+,D)P(G|P=yes)}{\sum_{S,G,D} P(S)P(R=+)P(D|S)P(P=no|R=+,D)P(G|P=no)} \\
&= \frac{\sum_{S,D} P(S)P(R=+)P(D|S)P(P=yes|R=+,D)}{\sum_{S,D} P(S)P(R=+)P(D|S)P(P=no|R=+,D)} \\
&= \frac{\sum_{S,D} P(S)P(D|S)P(P=yes|R=+,D)}{\sum_{S,D} P(S)P(D|S)P(P=no|R=+,D)}
\end{aligned}$$

where,

$$\begin{aligned}
&\sum_{S,D} P(S)P(D|S)P(P=yes|R=+,D) \\
&= P(S=yes)P(D=yes|S=yes)P(P=yes|R=+,D=yes) + \\
&\quad P(S=no)P(D=yes|S=no)P(P=yes|R=+,D=yes) + \\
&\quad P(S=yes)P(D=no|S=yes)P(P=yes|R=+,D=no) + \\
&\quad P(S=no)P(D=no|S=no)P(P=yes|R=+,D=no) \\
&= 0.3 \times 0.7 \times 0.7 + 0.7 \times 0.2 \times 0.7 + 0.3 \times 0.3 \times 0.3 + 0.7 \times 0.8 \times 0.3 \\
&= 0.44 \\
&\sum_{S,D} P(S)P(D|S)P(P=no|R=+,D) \\
&= P(S=yes)P(D=yes|S=yes)P(P=no|R=+,D=yes) + \\
&\quad P(S=no)P(D=yes|S=no)P(P=no|R=+,D=yes) + \\
&\quad P(S=yes)P(D=no|S=yes)P(P=no|R=+,D=no) + \\
&\quad P(S=no)P(D=no|S=no)P(P=no|R=+,D=no) \\
&= 0.3 \times 0.7 \times 0.3 + 0.7 \times 0.2 \times 0.3 + 0.3 \times 0.3 \times 0.7 + 0.7 \times 0.8 \times 0.7 \\
&= 0.56
\end{aligned}$$

Thus,

$$\frac{P(P=yes|R=+)}{P(P=no|R=+)} = \frac{P(P=yes,R=+)}{P(P=no,R=+)} = \frac{0.44}{0.56} < 1$$

So, the class should be no.

5. An e-commerce company has launched an online marketing campaign using banner advertisements placed on various third-party web sites to drive visitors to the company's web site. The web site has received 50,000 hits since the campaign was launched, out of which 10,000 of them were the result of click-throughs of the banner advertisement. Furthermore, 20% of the visitors directed to the Web site via the banner advertisement bought an item at the web site (compared to only 6% of the visitors arriving at the web site without clicking on a banner advertisement who ended up buying an item). Based on this information:

   (a) Calculate the probability that a visitor will buy an item at the website (regardless of how the visitor arrives at the website).
   Answer: Based on the information provided in the question, we have: P(Ad) = 10000/50000 = 0.2, P(Buy|Ad) = 0.2, and P(Buy|No Ad) = 0.06. Therefore,

$$\begin{aligned} \text{P(Buy)} &= \text{P(Buy, Ad)} + \text{P(Buy, No Ad)} \\ &= \text{P(Buy} \mid \text{Ad)P(Ad)} + \text{P(Buy} \mid \text{No Ad)P(No Ad)} \\ &= 0.2 \times 0.2 + 0.06 \times 0.8 \\ &= 0.088 \end{aligned}$$

   (b) Determine whether the advertisement campaign is successful. To do this, you need to check the following: among the visitors who made a purchase at the web site, are they more likely to arrive at the web site via the banner advertisement or arrive without clicking on the banner? Show your calculations clearly.
   Answer: We need to compare P(Ad|Buy) against P(No Ad|Buy):

$$\begin{aligned} \frac{\text{P(Ad|Buy)}}{\text{P(No Ad|Buy)}} &= \frac{\text{P(Buy|Ad) P(Ad)}}{\text{P(Buy|No Ad) P(No Ad)}} \\ &= \frac{0.2 \times 0.2}{0.06 \times 0.8} \\ &= 0.833 \end{aligned}$$

   Since the ratio is less than 1, the advertisement is not successful.

6. Consider the following training set for predicting whether there is traffic congestion in the morning on a highway for a particular day. There are 100 examples in the training set, with 40% positive (congestion) and 60% negative (no congestion) examples.

Accident	Weather	Construction	Number of positive training examples	Number of negative training examples
no	good	no	5	30
no	good	yes	10	20
yes	good	no	10	5
yes	bad	no	10	5
yes	bad	yes	5	0

(a) Compute the class conditional probabilities P(accident=yes|+), P(accident=yes|−), P(weather=good|+), P(weather=good|−), P(construction=yes|+), and P(construction=yes|−).

Answer:

P(accident=yes|+) = 25/40, P(accident=yes|−) = 10/60, P(weather=good|+) = 25/40, P(weather=good|−) = 55/60, P(construction=yes|+) = 15/40, P(construction=yes|−) = 20/60.

(b) Use the class conditional probabilities given in the previous question to predict the class label of a test example with the following feature set: (Accident = no, weather = bad, construction = yes) by applying the naïve Bayes classifier.

Answer: Let X = (Accident = no, weather = bad, construction = yes) be the test example. We need to compare $P(+|X)$ against $P(-|X)$, where the positive class represents congestion.

$$\begin{aligned}\frac{P(+|X)}{P(-|X)} &= \frac{P(X|+)P(+)}{P(X|-)P(-)} \\ &= \frac{\text{P(Accident=no|+)P(weather=bad|+)P(construction=yes|+)P(+)}}{\text{P(Accident=no|-)P(weather=bad|-)P(construction=yes|-)P(-)}} \\ &= \frac{15/40 \times 15/40 \times 15/40 \times 40/100}{50/60 \times 5/60 \times 20/60 \times 60/100} \\ &= 1.5188\end{aligned}$$

Since the ratio is greater than 1, $P(+|X) > P(-|X)$. Thus, it is classified as congestion.

7. Consider the problem of predicting whether there will be traffic congestion on a specific highway using the Bayesian network shown in Figure 4.4. Use the Bayesian network to answer the following questions.

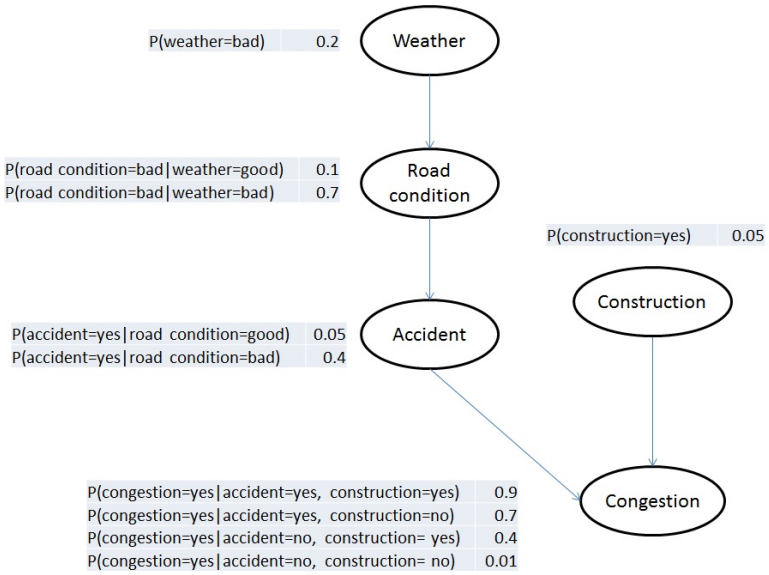


Figure 4.4. Bayesian network for question 7.

(a) Determine whether each of the following independence or conditional independence conditions hold according to the Bayesian network. To receive full credit, you must show your steps clearly (to prove/disprove the assumptions).

i. Accident $\perp$ Weather
Answer: Let A: Accident, W: weather, R: road condition, C: construction, y: congestion. We need to check whether $P(A, W) = P(A)P(W)$.

$$\begin{aligned} P(A,W) &= \sum_{R,C,y} P(A,W,R,C,y) \quad \text{(from sum rule)} \\ &= \sum_{R,C,y} P(y|A,C)P(C)P(A|R)P(R|W)P(W) \quad \text{(from DAG)} \\ &= \sum_{R,C} P(C)P(A|R)P(R|W)P(W) \\ &= \sum_{R} P(A|R)P(R|W)P(W) \end{aligned}$$

This is equal to $P(A)P(W)$ only if $\sum_R P(A|R)P(R|W) = P(A)$. Since this is not necessarily true (due to the second term $P(R|W)$), we cannot conclude that $P(A, W) = P(A)P(W)$ based on the directed acyclic graph. Thus, in general, Accident is not independent of Weather.

ii. Congestion $\perp$ Accident | Weather, Construction
Answer: We need to check whether $P(y, A|W, C) = P(y|W, C)P(A|W, C)$.

$$\begin{aligned}
P(y, A|W, C) &= \frac{P(y, A, W, C)}{P(W, C)} \\
&= \frac{\sum_R P(y, A, W, C, R)}{\sum_{y,A,R} P(y, A, W, C, R)} \\
&= \frac{\sum_R P(y|A, C)P(C)P(A|R)P(R|W)P(W)}{\sum_{y,A,R} P(y|A, C)P(C)P(A|R)P(R|W)P(W)} \\
&= \frac{\sum_R P(y|A, C)P(C)P(A|R)P(R|W)P(W)}{\sum_{A,R} P(C)P(A|R)P(R|W)P(W)} \\
&= \frac{\sum_R P(y|A, C)P(C)P(A|R)P(R|W)P(W)}{P(C)P(W)\sum_R P(R|W)} \\
&= \frac{P(y|A, C)P(C)P(W)\sum_R P(A|R)P(R|W)}{P(C)P(W)} \\
&= P(y|A, C)\sum_R P(A|R)P(R|W) \qquad (4.11)
\end{aligned}$$

where we have used the fact that $\sum_y P(y|A, C) = 1$, $\sum_A P(A|R) = 1$, and $\sum_R P(R|W) = 1$.

Furthermore, for the right-hand side:

$$
\begin{aligned}
P(y|W,C) &= \frac{P(y,W,C)}{P(W,C)} \\
&= \frac{\sum_{A,R} P(y,A,W,C,R)}{\sum_{y,A,R} P(y,A,W,C,R)} \\
&= \frac{\sum_{A,R} P(y|A,C)P(C)P(A|R)P(R|W)P(W)}{\sum_{y,A,R} P(y|A,C)P(C)P(A|R)P(R|W)P(W)} \\
&= \frac{\sum_{A,R} P(y|A,C)P(C)P(A|R)P(R|W)P(W)}{P(C)P(W)} \\
&= \sum_{A,R} P(y|A,C)P(A|R)P(R|W) \qquad (4.12) \\
P(A|W,C) &= \frac{P(A,W,C)}{P(W,C)} \\
&= \frac{\sum_{y,R} P(y,A,W,C,R)}{\sum_{y,A,R} P(y,A,W,C,R)} \\
&= \frac{\sum_{y,R} P(y|A,C)P(C)P(A|R)P(R|W)P(W)}{\sum_{y,A,R} P(y|A,C)P(C)P(A|R)P(R|W)P(W)} \\
&= \frac{P(C)P(W)\sum_{R} P(A|R)P(R|W)}{P(C)P(W)} \\
&= \sum_{R} P(A|R)P(R|W) \qquad (4.13)
\end{aligned}
$$

Plugging Equation (4.13) into (4.11), we have:

$$P(y,A|W,C) = P(y|A,C)P(A|W,C)$$

However, from Equation (4.12), since $P(y|W,C) \neq P(y|A,C)$,

$$P(y,A|W,C) \neq P(y|W,C)P(A|W,C),$$

which means, y and A are not conditionally independent given W and C.

(b) Suppose the weather is bad on a particular day and there is ongoing construction on the highway. Predict whether the highway will most likely be congested or not.

Answer: We need to compare $P(y = yes|W = bad, C = yes)$ against $P(y = no|W = bad, C = yes)$:

$$\begin{aligned}
& \frac{P(y = yes|W = bad, C = yes)}{P(y = no|W = bad, C = yes)} \\
= & \frac{P(y = yes, W = bad, C = yes)/P(W = bad, C = yes)}{P(y = no, W = bad, C = yes)/P(W = bad, C = yes)} \\
= & \frac{\sum_{A,R} P(y = yes, W = bad, C = yes, A, R)}{\sum_{A,R} P(y = no, W = bad, C = yes, A, R)} \\
= & \frac{\sum_{A,R} P(y = yes|A, C = yes)P(A|R)P(R|W = bad)P(C = yes)P(W = bad)}{\sum_{A,R} P(y = no|A, C = yes)P(A|R)P(R|W = bad)P(C = yes)P(W = bad)} \\
= & \frac{\sum_{A,R} P(y = yes|A, C = yes)P(A|R)P(R|W = bad)}{\sum_{A,R} P(y = no|A, C = yes)P(A|R)P(R|W = bad)}
\end{aligned}$$

For the numerator term:

$$\begin{aligned}
& \sum_{A,R} P(y = yes|A, C = yes)P(A|R)P(R|W = bad) \\
= & P(y = yes|A = yes, C = yes)P(A = yes|R = bad)P(R = bad|W = bad) \\
& +P(y = yes|A = yes, C = yes)P(A = yes|R = good)P(R = good|W = bad) \\
& +P(y = yes|A = no, C = yes)P(A = no|R = bad)P(R = bad|W = bad) \\
& +P(y = yes|A = no, C = yes)P(A = no|R = good)P(R = good|W = bad) \\
= & 0.9 \times 0.4 \times 0.7 + 0.9 \times 0.05 \times 0.3 + 0.4 \times 0.6 \times 0.7 + 0.4 \times 0.95 \times 0.3 \\
= & 0.5475
\end{aligned}$$

For the denominator term:

$$\begin{aligned}
& \sum_{A,R} P(y = no|A, C = yes)P(A|R)P(R|W = bad) \\
= & P(y = no|A = yes, C = yes)P(A = yes|R = bad)P(R = bad|W = bad) \\
& +P(y = no|A = yes, C = yes)P(A = yes|R = good)P(R = good|W = bad) \\
& +P(y = no|A = no, C = yes)P(A = no|R = bad)P(R = bad|W = bad) \\
& +P(y = no|A = no, C = yes)P(A = no|R = good)P(R = good|W = bad) \\
= & 0.1 \times 0.4 \times 0.7 + 0.1 \times 0.05 \times 0.3 + 0.6 \times 0.6 \times 0.7 + 0.6 \times 0.95 \times 0.3 \\
= & 0.4525
\end{aligned}$$

Since the numerator is larger than the denominator, we classify the highway as likely to be congested.

4.3 Ensemble Classifier

1. Consider the following loss function for the AdaBoost classifier:

$$\mathcal{L}_{\text{AdaBoost}} = \frac{1}{N}\sum_{i=1}^{N}\exp\Big[-y_i f(x_i)\Big],$$

where N is the number of training examples, $y_i \in \{-1, +1\}$ is the true class label, and $f(x_i) \in \{-1, +1\}$ is the predicted class label.

(a) The average misclassification error on training data is defined as:

$$\mathcal{L}_{\text{error}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\Big[y_i \neq f(x_i)\Big],$$

where $\mathcal{I}[\cdot]$ is an indicator function whose value is equal to 1 if its argument is true and 0 otherwise. Compare $\mathcal{L}_{\text{AdaBoost}}$ against $\mathcal{L}_{\text{error}}$ and find an inequality relation between them, i.e., is it possible to write one of them as greater than or equal to the other. Make sure you state your reason clearly.

Answer:

$$\begin{aligned}\mathcal{L}_{\text{error}} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\Big[y_i \neq f(x_i)\Big] \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\Big[y_i f(x_i) < 0\Big]\end{aligned}$$

If $y_i f(x_i) < 0$, then $\mathcal{I}[y_i f(x_i) < 0] = 1$ but $\exp[y_i f(x_i)] > 1$. Conversely, if $y_i f(x_i) > 0$, then $\mathcal{I}[y_i f(x_i) < 0] = 1$ but $\exp[y_i f(x_i)] > 0$. Therefore

$$\mathcal{L}_{\text{error}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\Big[y_i f(x_i) < 0\Big] < \frac{1}{N}\sum_{i=1}^{N}\exp\Big[y_i f(x_i)\Big] = \mathcal{L}_{\text{AdaBoost}}$$

(b) In AdaBoost, the partition function (i.e., normalization factor) in each boosting round j is given by

$$Z_j = \sum_{i=1}^{N} w_i^{(j)} \exp\Big[-y_i \alpha_j f(\mathbf{x}_i)\Big]$$

while the error in each round is given by

$$\epsilon_j = \sum_{i=1}^{N} w_i^{(j)} \mathcal{I}\Big[y_i \neq f(\mathbf{x}_i)\Big]$$

Write an expression for Z_j in terms of ϵ_j.
Answer:

$$\begin{aligned} Z_j &= \sum_{i=1}^{N} w_i^{(j)} \exp\Big[-y_i \alpha_j f(\mathbf{x}_i)\Big] \\ &= \sum_{i:y_i=f(\mathbf{x}_i)} w_i^{(j)} \exp\Big[-\alpha_j\Big] + \sum_{i:y_i\neq f(\mathbf{x}_i)} w_i^{(j)} \exp\Big[\alpha_j\Big] \\ &= (1-\epsilon_j)\exp\Big[-\alpha_j\Big] + \epsilon_j \exp\Big[\alpha_j\Big] \end{aligned} \tag{4.14}$$

where we have used the fact that

$$\epsilon_j = \sum_{i=1}^{N} w_i^{(j)} \mathcal{I}\Big[y_i \neq f(\mathbf{x}_i)\Big] = \sum_{i:y_i\neq f(\mathbf{x}_i)} w_i^{(j)}$$

and

$$\begin{aligned} 1-\epsilon_j &= \sum_{i=1}^{N} w_i^{(j)} - \sum_{i=1}^{N} w_i^{(j)} \mathcal{I}\Big[y_i \neq f(\mathbf{x}_i)\Big] \\ &= \sum_{i=1}^{N} w_i^{(j)} \Big(1 - \mathcal{I}\Big[y_i \neq f(\mathbf{x}_i)\Big]\Big) \\ &= \sum_{i:y_i=f(\mathbf{x}_i)} w_i^{(j)} \end{aligned}$$

Furthermore, it can be easily shown that $\alpha_j = \frac{1}{2}\log\frac{1-\epsilon_j}{\epsilon_j}$ by taking the partial derivative of Z_j with respect to α_j and setting it to zero.

Plugging this into Equation (4.14), we have:

$$\begin{aligned}
Z_j &= (1-\epsilon_j)\exp\left[-\frac{1}{2}\log\frac{1-\epsilon_j}{\epsilon_j}\right] + \epsilon_j \exp\left[\frac{1}{2}\log\frac{1-\epsilon_j}{\epsilon_j}\right] \\
&= (1-\epsilon_j)\left(\exp\left[\log\frac{1-\epsilon_j}{\epsilon_j}\right]\right)^{-\frac{1}{2}} + \epsilon_j\left(\exp\left[\log\frac{1-\epsilon_j}{\epsilon_j}\right]\right)^{\frac{1}{2}} \\
&= (1-\epsilon_j)\left(\frac{1-\epsilon_j}{\epsilon_j}\right)^{-\frac{1}{2}} + \epsilon_j\left(\frac{1-\epsilon_j}{\epsilon_j}\right)^{\frac{1}{2}} \\
&= (1-\epsilon_j)\sqrt{\frac{\epsilon_j}{1-\epsilon_j}} + \epsilon_j\sqrt{\frac{1-\epsilon_j}{\epsilon_j}} \\
&= 2\sqrt{\epsilon_j(1-\epsilon_j)}
\end{aligned}$$

(c) Consider the following 8 weighted training examples along with their true and predicted class labels after performing j rounds of boosting iterations:

Data point i	1	2	3	4	5	6	7	8
Weight, $w_i^{(j)}$	0.35	0.2	0.1	0.05	0.05	0.05	0.1	0.1
y_i	-1	+1	+1	+1	+1	-1	-1	-1
$f(x_i)$	+1	+1	+1	+1	-1	-1	-1	-1

i. Based on the given information, calculate ϵ_j and α_j.
Answer: In the example, only data points 1 and 5 are misclassified by $f(\mathbf{x})$. Thus

$$\epsilon_j = 0.35 + 0.05 = 0.40, \quad \alpha_j = \frac{1}{2}\log_e\frac{1-\epsilon_j}{\epsilon_j} = \frac{1}{2}\log_e\frac{0.6}{0.4} = 0.2027$$

ii. Show the new weights for each of the 8 training examples, $w^{(j+1)}$.
Answer:

i	1	2	3	4	5	6	7	8
$w_i^{(j)}$	0.35	0.2	0.1	0.05	0.05	0.05	0.1	0.1
y_i	-1	+1	+1	+1	+1	-1	-1	-1
$f(x_i)$	+1	+1	+1	+1	-1	-1	-1	-1
$w_i^{(j+1)}$	0.4375	0.1667	0.0833	0.0417	0.0625	0.0417	0.0833	0.0833

5

Association Analysis

5.1 Association Rules and Frequent Pattern Mining

1. Consider the following set of candidate 3-itemsets:

   $\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, c, f\}, \{a, d, f\},$
   $\{b, c, d\}, \{b, c, e\}, \{b, d, f\}, \{c, d, e\}, \{c, d, f\}.$

   (a) Construct a hash tree for storing the above 3-itemsets.

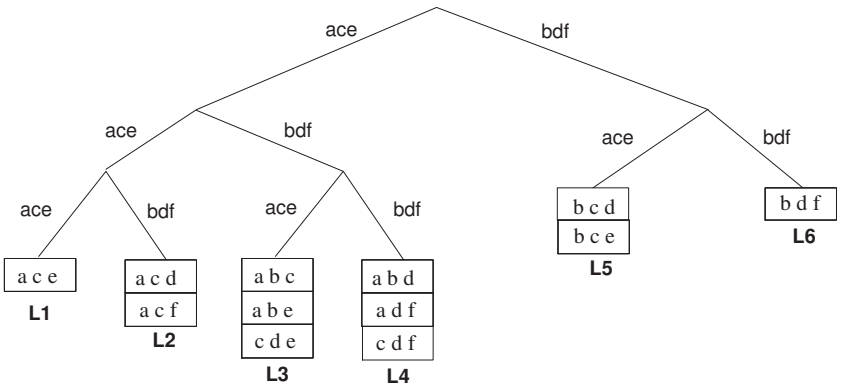


Figure 5.1. Hash tree.

   (b) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.

Answer: {abcd}, {abce}, {abde}, {acde}, {acdf}, {acef}, {bcde}, {cdef}.

(c) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.

Answer: {abcd}, {abce}, {acdf}. The rest of the itemsets are pruned because they contain at least one infrequent 3-itemsets.

(d) Based on the list of frequent 3-itemsets given above, is it possible to generate a frequent 5-itemset? State your reason clearly.

Answer: No. A frequent 5-itemset must contain at least 5 frequent 4-itemsets.

2. Consider a transaction dataset that contains five items, $\{A, B, C, D, E\}$.

(a) Suppose the support of $\{A, B\}$ is the same as the support of $\{A, B, C\}$, which one of the following statements are true:

i. Support of $\{A, B\}$ is the same as support of $\{A, C\}$.
Answer: False. As a counter-example, let $\{t_1, t_2, t_3\}$ be transactions that contain $\{A, B\}$ and $\{A, B, C\}$ but $\{t_1, t_2, t_3, t_4\}$ are transactions that contain $\{A, C\}$.

ii. The confidence of the rule $\{A, B\} \rightarrow \{C\}$ is 100%.
Answer: True because confidence of the rule is given by the ratio of support between $\{A, B, C\}$ and $\{A, B\}$.

iii. The support of $\{A, B, D\}$ is the same as the support of $\{B, C, D\}$.
Answer: False. As a counter-example, let $\{t_1, t_2, t_3\}$ be transactions that contain both $\{A, B\}$ and $\{A, B, C\}$, whereas $\{t_1, t_2, t_3, t_4\}$ be transactions that contain $\{B, C\}$. If D is contained in $\{t_1, t_2, t_4, t_5\}$, then support count of $\{A, B, D\}$ is 2 whereas support count of $\{B, C, D\}$ is 3.

iv. $\{A, B, D\}$ is not a closed itemset.
Answer: True because support of $\{A, B, D\}$ must be the same as support of $\{A, B, C, D\}$.

(b) Suppose the support of $\{A, B\}$ is the same as the support of $\{B, C\}$, which one of the following statements are true:

i. All transactions that contain $\{A, B\}$ also contain $\{B, C\}$.
Answer: False. As a counter-example, let $\{t_1, t_2, t_3, t_4\}$ be transactions that contain B, $\{t_1, t_2\}$ be transactions that contain A and $\{t_3, t_4\}$ be transactions that contain C.

ii. The confidence of the rule $\{A, B\} \rightarrow \{C\}$ is 100%.
Answer: False. As a counter-example, let $\{t_1, t_2, t_3, t_4\}$ be transactions that contain B, $\{t_1, t_2\}$ be transactions that contain A and $\{t_3, t_4\}$ be transactions that contain C. In this case, support of $\{A, B, C\}$ is 0, so confidence of the rule is also 0.

iii. The support of $\{A, B, D\}$ is the same as the support of $\{B, C, D\}$.
Answer: False. As a counter-example, let $\{t_1, t_2, t_3, t_4\}$ be transactions that contain B, $\{t_1, t_2\}$ be transactions that contain A and $\{t_3, t_4\}$ be transactions that contain C. Also, let $\{t_4, t_5\}$ be transactions that contain D. In this case, support count of $\{A, B, D\} = 0$ whereas support count of $\{B, C, D\}$ is 1.

iv. $\{A, B, D\}$ is not a closed itemset.
Answer: False. There is no guarantee that support of $\{A, B, D\}$ is identical to at least one of its supersets.

(c) Suppose all the transactions that contain $\{A, B\}$ also contain $\{B, C\}$, which one of the following statements are true:

i. The confidence of the rule $\{A, B\} \rightarrow \{C\}$ is 100%.
Answer: True because $s(\{A, B, C\}) = s(\{A, B\})$.

ii. The support of $\{A, B, D\}$ is the same as the support of $\{B, C, D\}$.
Answer: False because some transactions that contain $\{B, C\}$ may not necessarily contain $\{A, B\}$. If all the transactions that contain $\{B, C\}$ also contain D, then the support for $\{A, B, D\}$ may not be the same as support of $\{B, C, D\}$.

iii. $\{A, B, D\}$ is not a closed itemset.
Answer: True because support of $\{A, B, D\}$ is the same as the support of $\{A, B, C, D\}$.

(d) Suppose the rules $\{A, B\} \rightarrow C$ has the same confidence as $\{A, B\} \rightarrow D$, which one of the following statements are true:

i. The confidence of the $\{A, B\} \rightarrow \{C, D\}$ is the same as the confidence of $\{A, B\} \rightarrow \{C\}$.
Answer: False. Since the rules have identical confidence, therefore support of $\{A, B, C\}$ is the same as support of $\{A, B, D\}$ but the transactions that contain $\{A, B, C\}$ may not be the same as those that contain $\{A, B, D\}$.

ii. All transactions that contain {A,B,C} also contain {A,B,D}.
Answer: False. Since the rules have identical confidence, therefore support of $\{A, B, C\}$ is the same as support of $\{A, B, D\}$

but the transactions that contain $\{A, B, C\}$ may not be the same as those that contain $\{A, B, D\}$.

iii. $\{A, B, C\}$ is not a closed itemset.
Answer: False. Although support of $\{A, B, C\}$ is the same as support of $\{A, B, D\}$, there is no guarantee it is the same as support of $\{A, B, C, D\}$ since $\{A, B, C\}$ and $\{A, B, D\}$ might be contained in different transactions.

(e) Suppose we are interested to find all the **closed itemsets** in a given data set. For each of the following scenarios, list all the itemsets that are guaranteed to be not closed when:

i. Support of $\{B, C\}$ is equal to support of $\{A, B, C\}$.
Answer: $\{B, C\}$, $\{B, C, D\}$, $\{B, C, E\}$, and $\{B, C, D, E\}$.

ii. All the transactions that contain $\{A, D\}$ is a subset of transactions that contain $\{C\}$.
Answer: $\{A, D\}$, $\{A, B, D\}$, $\{A, D, E\}$, and $\{A, B, D, E\}$.

3. (a) What is the implication of setting your minimum confidence threshold lower than minimum support threshold?

Answer: Note that confidence of a rule $X \rightarrow Y$ cannot be less than its support, i.e., $P(X, Y)/P(X) \geq P(X, Y)$, since $P(X) \in (0, 1]$. As a result, all the rules derived from the frequent itemsets (i.e., whose support is greater than minimum support threshold) will pass the minimum confidence threshold, which means there is no confidence pruning. Furthermore, many of the rules of the form $X \rightarrow Y$ will have confidence values lower than their corresponding support of Y. Such rules tend to be spurious because they involve negatively correlated itemsets (see the tea-coffee example from the book).

(b) Consider an association rule $X \rightarrow Y$, where X and Y are itemsets. Let $P(X)$ denote the support of itemset X and $P(X, Y)$ denote the support of $X \cup Y$. Consider the following measure, which is known as the ϕ-coefficient:

$$\phi(X \rightarrow Y) = \frac{P(X, Y) - P(X)P(Y)}{\sqrt{P(X)P(Y)(1 - P(X))(1 - P(Y))}}$$

i. What is the ϕ-coefficient of association rules whose itemsets X and Y are independent.
Answer: When X and Y are independent, $P(X, Y) = P(X)P(Y)$. Therefore $\phi(X \rightarrow Y) = 0$.

ii. Is the measure monotone, anti-monotone, or non-monotone when the size of itemset X (i.e., left-hand side of the rule) is increased? In other words, when the set $X = \{a, b\}$ becomes $X' = \{a, b, c\}$, will the measure be non-decreasing (monotone), non-increasing (anti-monotone), or neither (non-monotone).
Answer: Non-monotone.

iii. Derive an expression for the upper bound of ϕ, called $\phi_{\max}$, which is a function defined in terms of $P(X)$ and $P(Y)$ only, i.e.:

$$\phi(X \rightarrow Y) \leq \phi_{\max}(X, Y), \quad \text{where } \phi_{\max}(X, Y) = f(P(X), P(Y))$$

Show that $\phi_{\max}$ is anti-monotone when the size of X increases (e.g., from $X = \{a, b\}$ to X'=$\{a, b, c\}$).
Answer:
Since $P(X, Y) \leq \min(P(X), P(Y))$, therefore:

$$\begin{aligned} \phi &\leq \frac{\min(P(X), P(Y)) - P(X)P(Y)}{\sqrt{P(X)P(Y)(1-P(X))(1-P(Y))}} \\ &= \begin{cases} \sqrt{\frac{P(X)(1-P(Y))}{P(Y)(1-P(X))}}, & \text{if } P(X) \leq P(Y); \\ \sqrt{\frac{P(Y)(1-P(X))}{P(X)(1-P(Y))}}, & \text{otherwise.} \end{cases} \\ &\equiv \phi_{\max} \end{aligned} \tag{5.1}$$

If $P(X) \leq P(Y)$, then increasing the size of X will only make $P(X)$ becomes smaller. So, $\phi_{\max}$ is anti-monotone.
If $P(X) > P(Y)$, then increasing the size of X may increase $\phi_{\max}$ as long as $P(X') > P(Y)$.

4. Consider the following set of frequent 2-itemsets:

$$\{p, q\}, \{p, r\}, \{p, s\}, \{p, t\}, \{q, r\}, \{q, t\}, \{r, s\}, \{s, t\}$$

(a) List all the candidate 3-itemsets produced during the candidate generation step of the Apriori algorithm.
Answer: {p,q,r}, {p,q,s}, {p,q,t}, {p,r,s}, {p,r,t}, {p,s,t}, {q,r,t}

(b) List all the candidate 3-itemsets that survive the pruning step of the Apriori algorithm.
Answer: {p,q,r}, {p,q,t}, {p,r,s}, {p,s,t}

(c) Based on the list of candidate 3-itemsets given above, is it possible to generate at least one frequent 4-itemset? State your reason clearly.

Answer: No because there are no viable candidate 4-itemset whose subsets of size-3 are all frequent. For example, even if all the candidate 3-itemsets in the previous question are frequent, the only candidate 4-itemset generated is {p,q,r,t}. Because {q,r,t} is not frequent, this candidate will be pruned.

5. Consider the transactions shown in the table below:

TID	Items
1	{a,b,c,d}
2	{a,c}
3	{b,c,e}
4	{b,d,e}
5	{a,c,d}
6	{a,b,d}
7	{b,e}
8	{a,b,c,e}
9	{a,b,c}
10	{a,b,d,e}

(a) Using the information given in the table above, draw a lattice structure of the itemsets. Label each node in the lattice with the following letter(s):

- **N**: If the itemset is not generated as a candidate itemset by the *Apriori* algorithm (using the candidate generation procedure described in class).
- **P**: If the itemset is generated as a candidate itemset by the *Apriori* algorithm but was subsequently pruned during the candidate pruning step.
- **I**: If the itemset is generated as a candidate itemset by the *Apriori* algorithm, survives the candidate pruning step, but fails the *minsup* threshold after determining its support from the transaction database.
- **F**: If the itemset is generated as a frequent itemset.
- **C**: If the itemset is closed.
- **M**: If the itemset is maximal frequent.

Assume that $minsup = 3$, which means an itemset must appear in at least 3 transactions to be frequent. A node can have more than one label. For example, if an itemset is maximal frequent, its corresponding node should have 3 labels, **F,C,M** because all maximal frequent itemsets are closed. A node can be closed but not frequent if its support is different than the support for all of its parents. You do not have to label the null node.

Answer: See Figure 5.7.

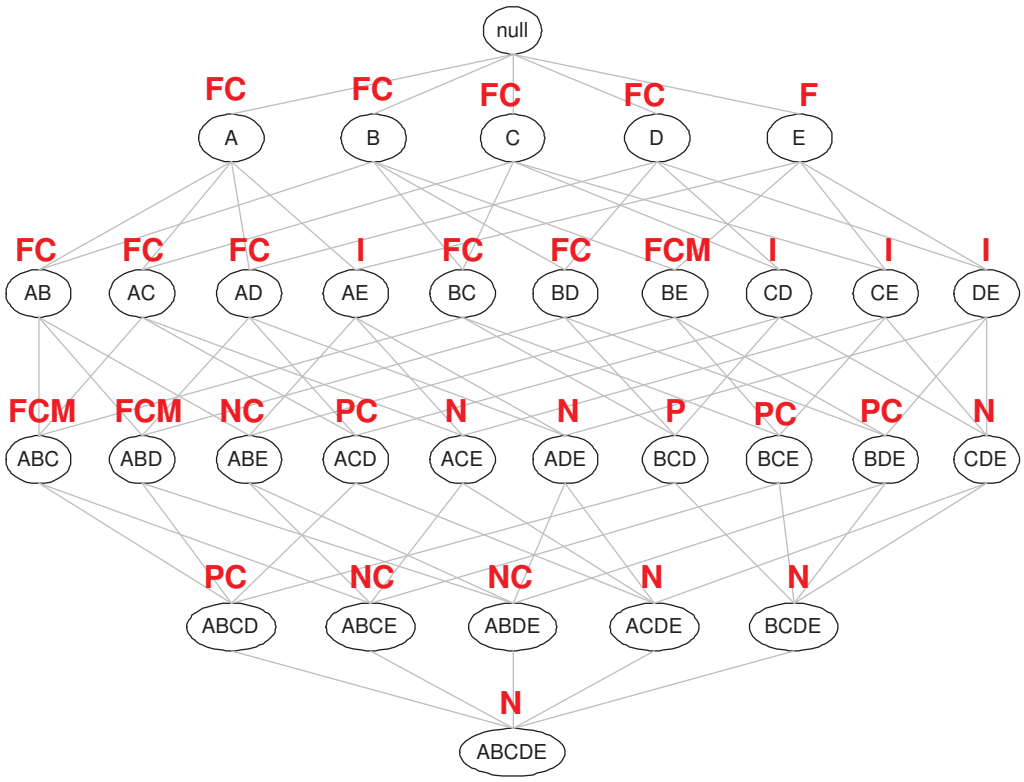


Figure 5.2. Search space for transaction database that contains 5 items.

(b) Calculate the pruning ratio of the *Apriori* algorithm when applied to the given dataset. Pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they are pruned during the candidate pruning step. In other words, since there are

31 possible itemsets (excluding null) in the lattice, pruning ratio is given by

$$\text{Pruning ratio} = \frac{\#\mathbf{N} + \#\mathbf{P}}{31},$$

where #**N** and #**P** are the number of nodes labeled as **N** and **P** in your lattice structure.

Answer: Pruning ratio = $\frac{14}{31}$.

6. Consider the following set of candidate 3-itemsets:

$\{p, q, r\}, \{p, q, s\}, \{p, q, t\}, \{p, r, s\}, \{p, r, t\}, \{q, r, s\}, \{q, r, t\},$
$\{q, s, t\}, \{r, s, t\}$.

(a) Construct a binary hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items p, r, t are hashed to the left child of a node, while items q, s, u are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf node irrespective of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemesets already stored at the leaf node is less than $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values.

Answer: See Figure 5.3.

(b) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.

Answer: {p,q,r,s}, {p,q,r,t}, {p,q,s,t}, {p,r,s,t}, {q,r,s,t}

(c) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.

Answer: {p,q,r,s}, {p,q,r,t}, {q,r,s,t}

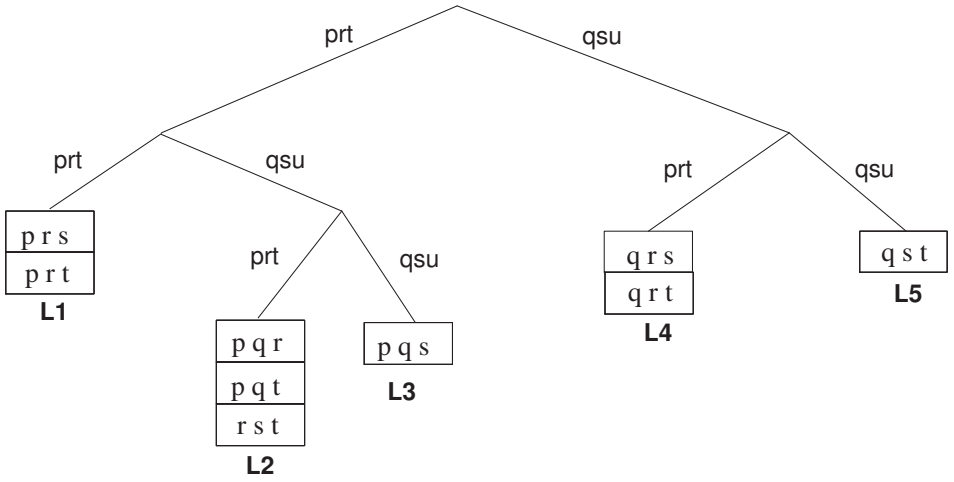


Figure 5.3. Hash tree for question 6.

(d) Based on the list of frequent 3-itemsets given above, is it possible to generate a frequent 5-itemset? State your reason clearly.
Answer: No. Even if all the candidate 4-itemsets given in the previous question are frequent, you still will not be able to generate any viable candidate 5-itemsets (since there are only 3 frequent 4-itemsets; any candidate 5-itemset must contain 5 frequent 4-itemsets).

7. Consider a transaction dataset that contains five items, $\{p, q, r, s, t\}$.

(a) Suppose the confidence of the rule $\{p, q\} \rightarrow \{s\}$ is 100%, which of the following statements are true:
i. All transactions that contain $\{p, q\}$ also contain $\{p, s\}$.
ii. The support counts of $\{p, q\}$, $\{p, s\}$, and $\{q, s\}$ are identical.
iii. $\{p, q\}$ is not a closed itemset.
iv. $\{p, q, s\}$ is not a closed itemset.

(b) Suppose the support of $\{p, q\}$ is identical to the support of $\{p, q, r\}$, which of the following statements are true:
i. Support of $\{p, q\}$ is the same as support of $\{p, r\}$.
ii. The confidence of the rule $\{p, q\} \rightarrow \{r\}$ is the same as the confidence of the rule $\{p, r\} \rightarrow \{q\}$.
iii. The support of $\{p, q, t\}$ is the same as the support of $\{q, r, t\}$.
iv. $\{p, q, t\}$ is not a closed itemset.

(c) Suppose all the transactions that contain $\{p, q\}$ also contain $\{p, s\}$, which of the following statements are true:

① The confidence of the rule $\{p, q\} \rightarrow \{s\}$ is 100%.

ii. The support of $\{p, q, t\}$ is the same as the support of $\{p, s, t\}$.

iii. The confidence of the rule $\{p, q\} \rightarrow \{s\}$ is the same as the confidence of $\{p, s\} \rightarrow \{q\}$

iv. $\{p, s, t\}$ is not a closed itemset.

(d) Suppose the rules $\{p, q\} \rightarrow \{r\}$ has the same confidence as $\{p, r\} \rightarrow \{q\}$, which one of the following statements are true:

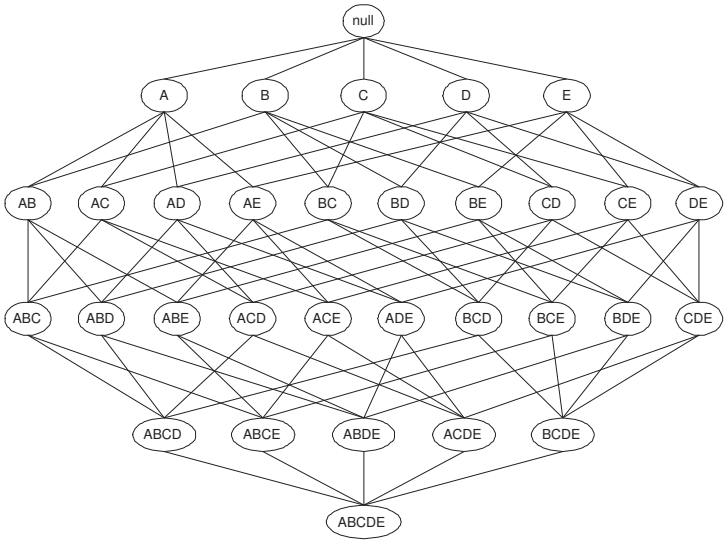
① The support of $\{p, q\}$ is the same as the support of $\{p, r\}$.

ii. All transactions that contain {p,q,t} also contain {r}.

iii. $\{p, q\}$ is not a closed itemset.

iv. $\{p, q, t\}$ is not a closed itemset.

8. Consider the lattice structure shown below.



We are interested in finding all **closed itemsets** in a given data set. For each of the following scenarios, list all the itemsets that are guaranteed to be not closed when:

(a) Support of $\{A\}$ is equal to support of $\{A, B, C\}$.

Answer: Any itemset $\mathcal{X}$ that contains {A} but not BOTH B and C are guaranteed to be not closed because the support of $\mathcal{X}$ is identical to the support of $\mathcal{X} \cup$ {B,C}. The answer to this question is: {A}, {A,B}, {A,C}, {A,D}, {A,E}, {A,B,D}, {A,B,E}, {A,C,D}, {A,C,E}, {A,D,E}, {A,B,D,E}, {A,C,D,E}.

(b) All the transactions that contain $\{A\}$ is a subset of transactions that contain $\{B\}$.

Answer: Any itemset $\mathcal{X}$ that contains {A} but not B are guaranteed to be not closed because the support of $\mathcal{X}$ is identical to the support of $\mathcal{X} \cup$ {B}. The answer to this question is: {A}, {A,C}, {A,D}, {A,E}, {A,C,D}, {A,C,E}, {A,D,E}, {A,C,D,E}.

9. Consider the following set of candidate 3-itemsets:

$$\{p,q,r\}, \{p,q,s\}, \{p,q,t\}, \{p,r,s\}, \{p,r,t\}, \{p,r,u\}, \{p,s,t\},$$
$$\{q,r,s\}, \{q,r,t\}, \{q,r,u\}, \{q,s,t\}, \{r,s,t\}, \{s,t,u\}$$

(a) Construct a hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items a, d are hashed to the left child of a node, items b, e are hashed to the middle child, while items c, f are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf node irrespective of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemsets stored at the leaf node is less than or equal to $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values.

Answer:

The hash tree is shown in Figure 5.4

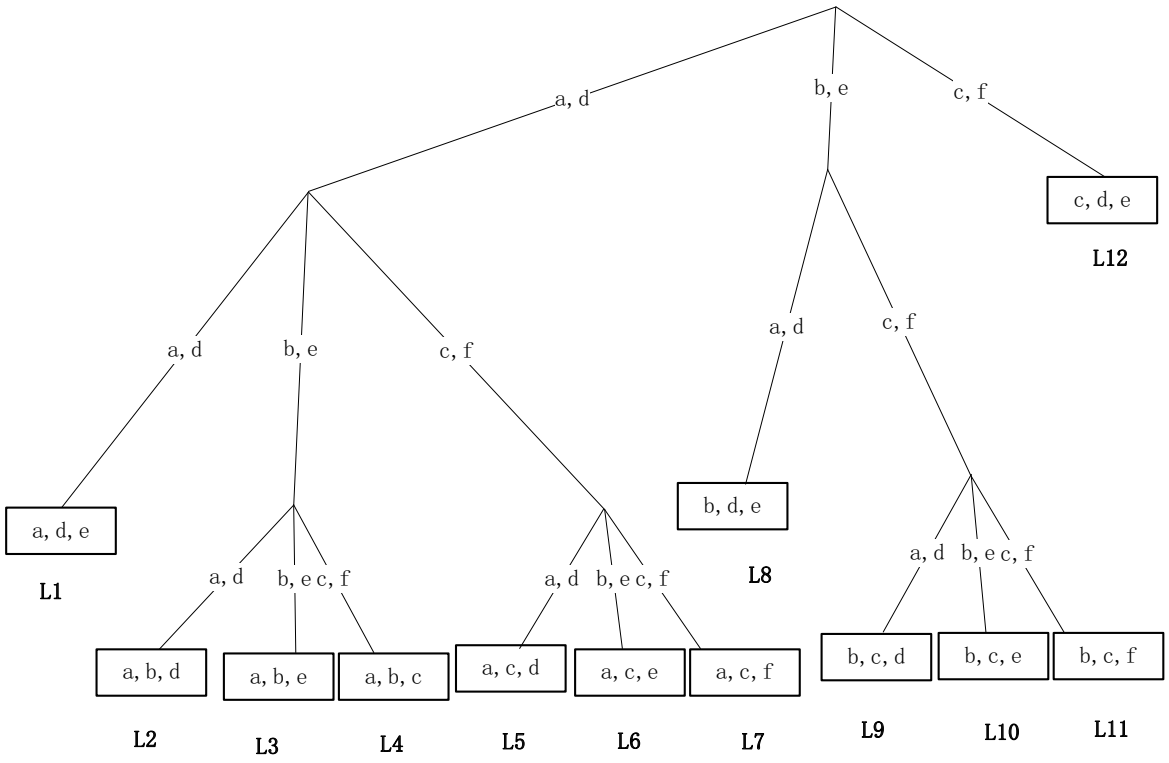


Figure 5.4. Hash tree for question 9(a).

(b) Consider a transaction that contains items {a,b,d,e,f}. Count the number of leaf nodes in the hash tree to which the transaction will be hashed into.

Answer:

There are 5 nodes. L1, L2, L3, L4, and L8

(c) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.

Answer:

{a,b,c,d}, {a,b,c,e}, {a,b,d,e}, {a,c,d,e}, {a,c,d,f}, {a,c,e,f}, {b,c,d,e}, {b,c,d,f}, {b,c,e,f}

(d) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.

Answer:

{a,b,c,d}, {a,b,c,e}, {a,b,d,e}, {a,c,d,e}, {b,c,d,e}

(e) Based on the list of frequent 3-itemsets given above, is it possible to generate a frequent 5-itemset? State your reason clearly.

Answer:

The candidate 5-itemset is {a,b,c,d,e}. We can see all the subsets of {a,b,c,d,e} are frequent. Hence, {a,b,c,d,e} is a frequent 5-itemset.

10. Consider a medical database $\mathcal{D}$ that contains N "transactions" and d "items".

(a) Suppose we are interested in rules of the form $X \rightarrow y$, where X is a k-itemset ($k \geq 1$) and y is an item (i.e., the right-hand side contains 1-itemsets only). These rules are known as discriminative rules. For medical diagnosis, such rules can be used to identify segments of the population who are susceptible to certain diseases. An example of such a rule is {male, age>60, alcohol-abuse} $\rightarrow$ {prostate-cancer}.

i. Count the maximum number of discriminative rules that can be extracted from the database $\mathcal{D}$ (in terms of d)

Answer:

$d(2^{d-1} - 1)$

ii. Count the maximum number of discriminative rules that can be extracted from a given frequent k-itemset X.

Answer:

k

iii. Suppose we are interested in finding discriminative rules whose support $\geq minsup$ and confidence $\geq$ minconf thresholds. Let X be a frequent k-itemset and X_{-1}, X_{-2}, and X_{-k} be the corresponding set of frequent $(k-1)$-itemsets after removing one of the items from X. What minimum support the $(k-1)$-itemsets must have in order to guarantee that the itemset X will generate at least one discriminative rule with confidence $\geq minconf$. Express your answer in terms of s_X (support of X) and $minconf$.

Answer:

The largest confidence of the discriminative rules generated from X is

$$maxconf = \max_i \frac{s_X}{s_{X_{-i}}}, i = 1, ...k$$

We want to find the minimum support that

$$\frac{S_X}{minsup(X_i)} \geq \max_i \frac{s_X}{s_{X-i}} \geq minconf$$

Hence,

$$minsup \leq \frac{s_X}{minconf}$$

(b) Suppose we are interested in rules of the form $y \rightarrow X$, where X is a k-itemset ($k \geq 1$) and y is an item (i.e., the left-hand side contains 1-itemsets only). These are also known as characteristic rules. Such rules can be used to identify a disease based on its symptoms. An example of such a rule is {pneumonia} $\rightarrow$ {cough, fever, shortness of breath}.

i. Count the maximum number of characteristic rules that can be extracted from the database $\mathcal{D}$ (in terms of d).
Answer:

$$d(2^{d-1} - 1)$$

ii. Count the maximum number of characteristic rules that can be extracted from a given frequent k-itemset X.
Answer: k

iii. Suppose we are interested in finding characteristic rules whose support $\geq minsup$ and confidence $\geq$ minconf thresholds. Design an Apriori-like algorithm for extracting such rules. In addition to the regular support and confidence pruning, your algorithm should take advantage of the support ordering of the items to further prune the search space of candidate itemsets and rules. Support ordering means the d items $\{x_1, x_2, \cdots x_d\}$ are ordered in increasing support order, i.e., $\forall i : \text{support}(x_i) \leq \text{support}(x_{i+1})$.
Hint: Find an expression for the upper bound confidence value for all characteristic rules that can be generated from a k-itemset X. Use the upper bound to prune candidate itemsets and candidate rules that are guaranteed to be of low confidence.
Answer:
The upper bound of confidence value for an ordered d itemset $x_1, x_2, ..., x_d$ is $\frac{s_X}{s_{x_1}}$. We can design our algorithm as Algorithm 5.3:

Algorithm 5.1 `Apriori`

```
1: F_k: frequent k-itemsets
2: L_k: candidate k-itemsets
3: U(X): upper bound of confidence value for rules generated from itemset
   X
4: R: Characteristic rule set
5: Let k = 1
6: Generate F_1=frequent 1-itemsets:
7: repeat
8:    Candidate Generation: Generate L_{k+1} from F_k
9:    Candidate Pruning:
10:   for each X in L_{k+1} do
11:      if s_X < minsup then
12:         Prune X
13:      else if U(X) < minconf then
14:         Prune X
15:      else
16:         Generate characteristic rule set R_{k+1} from X
17:         Prune off the rules whose confidence is less than minconf
18:         R ← R + R_{k+1}
19:      end if
20:   end for
21: until F_k is empty
22: return R
```

11. Consider the lattice structure shown in Figure 5.5.

    We are interested in finding all **maximal and closed itemsets** in a given data set.

    (a) What is the minimum and maximum number of maximal frequent itemsets one can generate from such a database?

    Answer: Minimum number is 0, maximum number is 10.

    (b) What is the minimum and maximum number of closed frequent itemsets one can generate from such a database?

    Answer: Minimum number is 0, maximum number is $2^5 - 1$.

    (c) If all the transactions that contain itemset $\{A, B\}$ also contain items C and E, list all the itemsets that are guaranteed to be not closed.

    Answer:

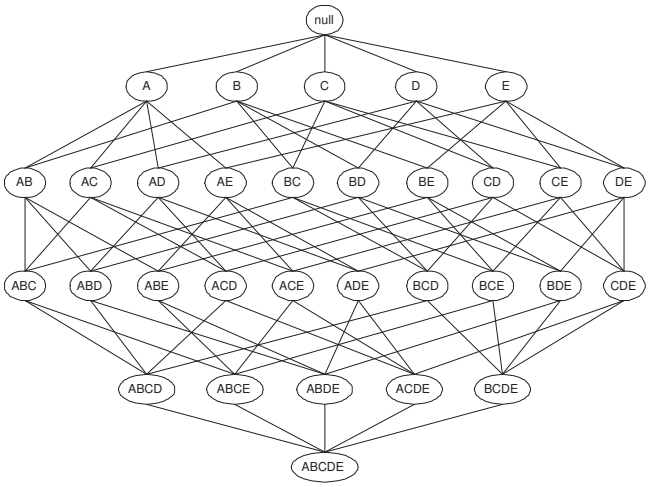


Figure 5.5. Search space for transaction database that contains 5 items.

{A,B}, {A,B,C}, {A,B,E}, {A,B,D}, {A,B,C,D}, {A,B,D,E}

(d) If the support of itemsets $\{B\}$ and $\{B, C, D\}$ are identical, list all the itemsets that are guaranteed to be not closed.

Answer:

{B}, {B,C}, {B,D}, {A,B}, {B,E}, {A,B,E}, {A,B,C}, {B,C,E}, {A,B,C,E}, {A,B,D}, {B,D,E}, {A,B,D,E}

12. Consider the contingency tables shown below for 3 pairs of items: (bread, milk), (pepsi, coke), and (caviar, wine).

	milk	$\overline{\text{milk}}$
bread	80	120
$\overline{\text{bread}}$	20	30

	coke	$\overline{\text{coke}}$
pepsi	20	80
$\overline{\text{pepsi}}$	100	50

	wine	$\overline{\text{wine}}$
caviar	15	5
$\overline{\text{caviar}}$	20	210

(a) Rank the following six rules (in increasing magnitude): bread $\rightarrow$ milk, milk $\rightarrow$ bread, coke $\rightarrow$ pepsi, pepsi $\rightarrow$ coke, wine $\rightarrow$ caviar, and caviar $\rightarrow$ wine according to the following measures: support, confidence, interest, and odds ratio.

Answer:

Rank according to support: wine $\rightarrow$ caviar, caviar $\rightarrow$ wine, coke $\rightarrow$ pepsi, pepsi $\rightarrow$ coke, bread $\rightarrow$ milk, milk $\rightarrow$ bread.

Rank according to confidence: coke $\rightarrow$ pepsi, pepsi $\rightarrow$ coke, bread $\rightarrow$ milk, wine $\rightarrow$ caviar, caviar $\rightarrow$ wine, milk $\rightarrow$ bread.

Rank according to interest: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

Rank according to odds ratio: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

(b) One difficulty in comparing the rules is that each item has different support counts (i.e., marginal totals). One way to avoid this difficulty is to standardize the tables to have uniform marginal totals. The effect of table standardization is shown below:

	B	$\overline{\text{B}}$	
A	f_{11}	f_{10}	f_{1+}
$\overline{\text{A}}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

$\longrightarrow$

	B	$\overline{\text{B}}$	
A	f^*_{11}	f^*_{10}	f^*_{1+}
$\overline{\text{A}}$	f^*_{01}	f^*_{00}	f^*_{0+}
	f^*_{+1}	f^*_{+0}	N

One popular method to standardize a contingency table is to apply the iterative proportional fitting (IPF) algorithm. Given an initial contingency table, the algorithm would iteratively modify all the cell entries in the table until the desired row and column sums are obtained. For equal marginal totals, you can set $f^*_{1+} = f^*_{0+} = f^*_{+1} = f^*_{+0} = N/2$.

Algorithm 5.2 `Iterative Proportional Fitting`

Input: f: 2×2 contingency matrix
Output: f: standardized contingency matrix

Set $f^*_{1+} = f^*_{0+} = f^*_{+1} = f^*_{+0} = N/2$
for k=1 to maxiter **do**
  if k is odd **then**
    Rescale each entry in f by row sum: $f^{(k)}_{ij} \leftarrow f^{(k-1)}_{ij} \times \frac{f^*_{i+}}{f^{(k-1)}_{i+}}$
  else
    Rescale each entry in f by column sum: $f^{(k)}_{ij} \leftarrow f^{(k-1)}_{ij} \times \frac{f^*_{+j}}{f^{(k-1)}_{+j}}$
  end if
end for
return f

Apply the IPF procedure to all three contingency tables (you can download the Matlab script ipf.m from the class web site or write the code yourself). Show the values of the contingency tables after

standardization. Rank the three tables in increasing order of magnitude using support, confidence, interest factor, and odds ratio.

Answer:

The contingency tables after the IPF procedure are:

	milk	$\overline{\text{milk}}$
bread	62.5	62.5
$\overline{\text{bread}}$	62.5	62.5

	coke	$\overline{\text{coke}}$
pepsi	32.65	92.35
$\overline{\text{pepsi}}$	92.35	32.65

	wine	$\overline{\text{wine}}$
caviar	106.10	18.90
$\overline{\text{caviar}}$	18.90	106.10

Using new contingency tables, the orders are shown in the following:

Rank according to support: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

Rank according to confidence: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

Rank according to interest: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

Rank according to odds ratio: coke → pepsi, pepsi → coke, bread → milk, milk → bread, wine → caviar, caviar → wine.

13. A clique is a pattern that contains highly similar items. If an itemset $X = \{x_1, x_2, \cdots x_k\}$ is a clique, then the presence of any item $x_i \in X$ in a transaction t implies the presence of all other items $x_j \in X - \{x_i\}$ in the transaction t with high probability. For example, if the words {`graph`, `node`, `link`} form a clique pattern in a set of documents, then any document that contains the word `graph` is highly likely to contain the words `node` and `link` as well[1]. For this exercise, you need to design an *Apriori*-like algorithm for efficient mining of such patterns.

    (a) Which of the following evaluation measures M for a given itemset $X = \{x_1, x_2, \cdots x_k\}$ is most appropriate to define clique patterns? An itemset X is considered to be a clique only if $M(X) \geq \tau$, where τ is a user-defined threshold. State your reason clearly.

    - $M(X)$ is the maximum support for one of the items in X, i.e., $M(X) = \max_{x_i \in X} s(x_i)$, where $s(x_i)$ is the support of x_i.
    - $M(X)$ is the minimum support for one of the items in X, i.e., $M(X) = \min_{x_i \in X} s(x_i)$, where $s(x_i)$ is the support of x_i.

[1] In this example, an item corresponds to a word and a transaction corresponds to a document.

- $M(X)$ is the maximum confidence of all association rules extracted from X, i.e., $M(X) = \max_{L \subset X} c(L \rightarrow X - L)$. For example, if $X = \{\texttt{graph}, \texttt{node}, \texttt{link}\}$, then $M(X)$ is the maximum confidence is computed from the rules $\{\texttt{graph}\} \rightarrow \{\texttt{node}, \texttt{link}\}$,$\{\texttt{node}\} \rightarrow \{\texttt{graph}, \texttt{link}\}$, $\{\texttt{node}, \texttt{graph}\} \rightarrow \{\texttt{link}\}$, etc.
- $M(X)$ is the minimum confidence of all association rules extracted from X, i.e., $M(X) = \min_{L \subset X} c(L \rightarrow X - L)$. For example, if $X = \{\texttt{graph}, \texttt{node}, \texttt{link}\}$, then $M(X)$ is the minimum confidence is computed from the rules $\{\texttt{graph}\} \rightarrow \{\texttt{node}, \texttt{link}\}$,$\{\texttt{node}\} \rightarrow \{\texttt{graph}, \texttt{link}\}$, $\{\texttt{node}, \texttt{graph}\} \rightarrow \{\texttt{link}\}$, etc.

Answer: $M(X) = \min_{L \subset X} c(L \rightarrow X - L)$ is the most appropriate measure for clique patterns. The measure ensures that if a subset of the itemset is present in an itemset X, the rest of the items in the itemset are likely to be present with a high probability (if τ is chosen appropriately).

(b) Check whether the "clique" measure found in the preceding question has anti-monotone property. A measure M is anti-monotone if $\forall X, z : M(X) \geq M(X \cup \{z\})$, i.e., M is non-increasing when the itemset X is expanded by adding another item z to the set.

Answer: Yes. The previous measure has an anti-monotone property. Let $X = \{x_1, x_2, \cdots, x_k\}$ and $\hat{X} = \{x_1, x_2, \cdots, x_k, z\}$ be an extension of X. Based on the definition of the measure $M(X)$:

$$\begin{aligned} M(X) &= \min_{L \subset X} c(L \rightarrow X - L) = \min_{L \subset X} \frac{P(X)}{P(L)} \\ &= \frac{P(X)}{\max_{L \subset X} P(L)} \\ &= \frac{P(X)}{\max_{x_j \in X} P(x_j)} \end{aligned} \tag{5.2}$$

where we have used the fact that $\max_{x_j \in X} P(x_j) \geq \max_{L \subset X} P(L)$ due to the anti-monotone property of the support measure. Thus,

$$M(\hat{X}) = \frac{P(X \cup \{z\})}{\max_{x_j \in X \cup \{z\}} P(x_j)} \tag{5.3}$$

Comparing the numerators and denominators of Equations (5.2) and (5.3), it is easy to see that

$$P(X \cup \{z\}) \leq P(X), \qquad \max_{x_j \in X \cup \{z\}} P(x_j) \geq \max_{x_j \in X} P(x_j)$$

The first inequality is due to the anti-monotone property of support measure. The second inequality follows since the maximum support of an item in an itemset will either increase or stay the same if we add another item into the set. Thus, $M(\hat{X}) \leq M(X)$, which means $M(X)$ has anti-monotone property.

(c) Design an *Apriori*-like algorithm to efficiently extract the clique patterns. The pseudo-code for the algorithm should look like this:

Algorithm 5.3 `Clique Finding Algorithm`

1: Let $F_1 = \{i \in I\}$, i.e., assume every 1-itemset is a clique pattern.
2: Count the support of each itemset in F_1.
3: Let $k = 2$
4: **repeat**
5: Candidate Generation
6: Candidate Pruning
7: Support Counting
8: Candidate Elimination
9: $k \leftarrow k + 1$
10: **until** F_k is empty
11: `return` $R = F_1 \cup F_2 \cup \cdots$, i.e. the set of all clique patterns.

State which of the following steps of the Apriori algorithm (candidate generation, candidate pruning, support counting, and candidate elimination) should be modified so that the output consists only of clique patterns. Explain clearly what type of modification you need to make for those steps. For example, if the candidate pruning step has to be modified, how would you do it?

Answer: The candidate generation step should be modified. Assume all items within an itemset are ordered in increasing support values, i.e., if $X = \{a, c, d\}$, then $P(a) \leq P(c) \leq P(d)$. Given a k-clique pattern $X = \{x_1, x_2, \cdots, x_{k-1}, x_k\}$, we merge it with another k-clique pattern $X' = \{x'_1, x'_2, \cdots, x'_{k-1}, x'_k\}$ to produce a candidate $(k+1)$ pattern $X \cup \{x'_k\}$ if the following two conditions are satisfied:

i. They share the first $k-1$ items in common, i.e., $x'_1 = x_1$, $x'_2 = x_2$, $\cdots$, $x'_{k-1} = x_{k-1}$ (but $x'_k \neq x_k$).
ii. $P(x'_k) \geq P(x_k)$ and $P(X)/P(x'_k) \geq \tau$.

The first condition is exactly the same as the candidate generation step for Apriori. The second condition ensures that the itemset $\{x_1, x_2, \cdots, x_k, x'_k\}$ is a viable candidate. If $P(X)/P(x'_k) < \tau$, then $P(X \cup \{x'\})/P(x'_k) < \tau$; so the itemset is guaranteed to be a non-clique pattern. The candidate pruning and support counting steps are exactly the same as the Apriori algorithm. The candidate elimination step is modified to prune all itemsets whose measure ($M(X)$) is below the user-specified threshold τ.

14. Consider the following set of candidate 3-itemsets:

$$\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, c, f\}, \{a, d, e\},$$
$$\{b, c, d\}, \{b, c, e\}, \{b, c, f\}, \{b, d, e\}, \{b, e, f\}, \{c, d, e\}, \{c, e, f\}$$

(a) Construct a hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items a, d are hashed to the left child of a node, items b, e are hashed to the middle child, while items c, f are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf
node irrespective of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemsets currently
stored at the leaf node is less than $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values. Label the leaf nodes (from left to right) as L_1, L_2, L_3, and so on.

Answer: Shown in Figure 5.6.

(b) Consider a transaction that contains items {a,b,d,e,f}. List all the leaf nodes in the hash tree to which the transaction will be hashed into.

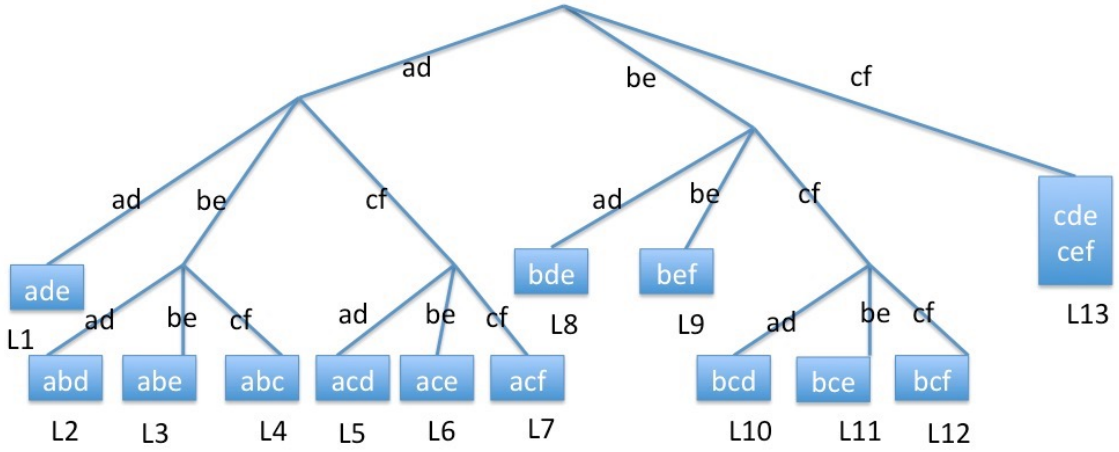


Figure 5.6. Hash tree.

Answer: Shown in Figure 5.4, the transaction will be hashed into leaf nodes L1, L2, L3, L4, L8 and L9 .

(c) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.
Answer: {a,b,c,d}, {a,b,c,e}, {a,b,d,e}, {a,c,d,e}, {a,c,d,f}, {a,c,e,f}, {b,c,d,e}, {b,c,d,f}, {b,c,e,f}.

(d) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.
Answer: {a,b,c,d}, {a,b,c,e}, {a,b,d,e}, {a,c,d,e}, {b,c,d,e}, {b,c,e,f}.

(e) If all the candidate 4-itemsets in part (d) are frequent, is it possible to generate a candidate 5-itemset? If yes, what is the candidate 5-itemset?
Answer: The candidate 5-itemset is {a,b,c,d,e}. We can see all the subsets of {a,b,c,d,e} are frequent. Hence, {a,b,c,d,e} is a frequent 5-itemset.

15. Consider the closing prices for five stocks (A, B, C, D, and E) listed in Table 5.1. Suppose you are interested in applying association rule mining to the data.

Table 5.1. Example of stock market data

Day	A	B	C	D	E
1	10.50	11.00	20.00	80.00	95.00
2	12.30	10.40	26.40	76.50	90.20
3	12.00	10.80	26.50	75.50	91.00
4	11.20	10.00	25.50	72.00	87.10
5	11.30	10.20	25.30	73.20	88.90
6	12.50	10.70	27.50	70.00	88.50
7	13.00	10.80	28.80	72.00	90.20
8	13.80	11.00	29.80	71.80	91.00
9	12.95	10.80	27.90	71.00	91.20
10	12.05	10.10	26.10	72.60	92.80
11	11.40	10.05	24.95	70.40	90.10

(a) Convert the stock market prices into transaction data. For each stock X on trading day t, compute the change in its closing price,

$$\Delta_X(t) = \frac{p_t(X) - p_{t-1}(X)}{p_{t-1}(X)},$$

where $p_t(X)$ is the price of stock X on day t. Next, create an "item" X-UP for trading day t if $\Delta_X(t) \geq 0.05$ (i.e., if the closing price is up by at least 5%) or X-DOWN if $\Delta_X(t) \leq -0.05$ (i.e., if the closing price is down by at least 5%). Assuming each transaction corresponds to a trading day (starting from Day 2), list all the 10 transactions (including its items) created from the data set. Note that there are 10 possible items: A-UP, A-DOWN, B-UP, B-DOWN, $\cdots$, E-UP, E-DOWN, that can appear in the transaction data.

Answer1:

Transaction1: {A-UP, B-DOWN, C-UP, E-DOWN};
Transaction2: {};
Transaction3: {A-DOWN, B-DOWN};
Transaction4: {};
Transaction5: {A-UP, C-UP };
Transaction6: {};
Transaction7: {A-UP};

Transaction8: {A-DOWN, C-DOWN };
Transaction9: {A-DOWN, B-DOWN, C-DOWN };
Transaction10: {A-DOWN };

Answer2:(Another acceptable answer)
Transaction1: {A-UP, B-DOWN, C-UP, E-DOWN};
Transaction2: {A-DOWN, B-DOWN};
Transaction3: {A-UP, C-UP };
Transaction4: {A-UP};
Transaction5: {A-DOWN, C-DOWN };
Transaction6: {A-DOWN, B-DOWN, C-DOWN };
Transaction7: {A-DOWN };

(b) Assuming the minimum support threshold is 20%, i.e., an itemset has to appear at least twice in the transaction data to be considered frequent, list all the frequent 1-itemsets, 2-itemsets, and so on (including their support values), that can be extracted from the data.
Answer1:
Frequent 1-itemsets: {A-UP} (0.3), {A-DOWN}(0.4), {B-DOWN} (0.3), {C-UP} (0.2), {C-DOWN} (0.2);
Frequent 2-itemsets: {A-DOWN, B-DOWN}(0.2), {A-DOWN, C-DOWN}(0.2), {A-UP, C-UP}(0.2).
Answer2:
Frequent 1-itemsets: {A-UP} (3/7), {A-DOWN}(4/7), {B-DOWN} (3/7), {C-UP} (2/7), {C-DOWN} (2/7);
Frequent 2-itemsets: {A-DOWN, B-DOWN}(2/7), {A-DOWN, C-DOWN}(2/7), {A-UP, C-UP}(2/7).

(c) Based on the frequent itemsets found in part (b), generate all the association rules with minsup = 20% and minconf = 60%. Ignore the rules in which their left or right hand side correspond to an empty set.
Answer1:
A-UP $\rightarrow$ C-UP: support = 0.2, confidence = 2/3;
B-DOWN $\rightarrow$ A-DOWN: support = 0.2, confidence = 2/3;
C-UP $\rightarrow$ A-UP: support = 0.2, confidence = 1;
C-DOWN $\rightarrow$ A-DOWN: support = 0.2, confidence = 1;

Answer2:
A-UP $\rightarrow$ C-UP: support = 2/7, confidence = 2/3;

B-DOWN → A-DOWN: support = 2/7, confidence = 2/3;
C-UP → A-UP: support = 2/7, confidence = 1;
C-DOWN → A-DOWN: support = 2/7, confidence = 1;

16. Consider the lattice diagram shown in Figure 5.7. Assume all the itemsets in the lattice are frequent.

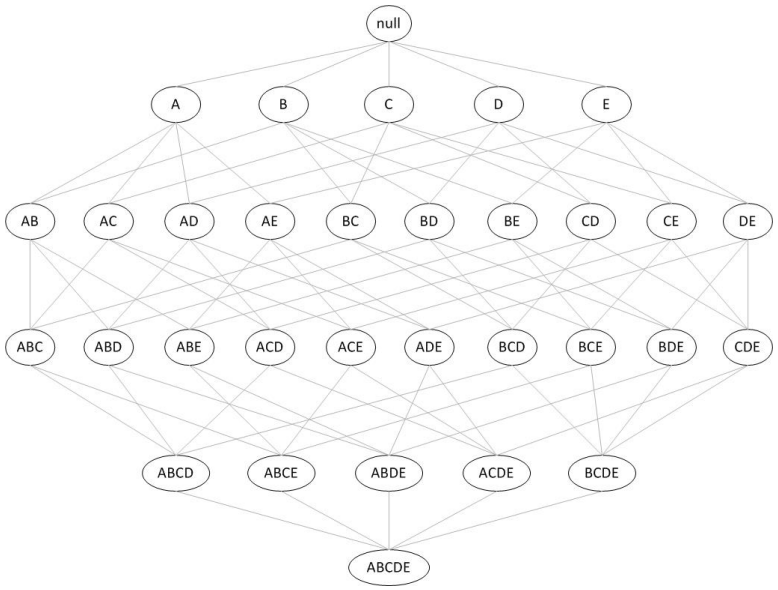


Figure 5.7. Search space for transaction database that contains 6 items.

List all the itemsets that are guaranteed to be **not closed** given each of the conditions below. We are interested in identifying the **non-closed itemsets** because they can be discarded by the closed itemset generation algorithm. Answer `none` if no itemsets can be guaranteed to be not closed by the given condition. Note that for some conditions, there may be more than one non-closed frequent itemset.

(a) Support of $\{A, B\}$ is equal to support of $\{A, B, D\}$
Answer: {A,B}, {A,B,C}, {A,B,E}, {A,B,C,E}.

(b) The transactions that contain $\{A, B\}$ is a subset of the transactions that contain $\{D\}$.
Answer: {A,B}, {A,B,C}, {A,B,E}, {A,B,C,E}.

(c) The confidence of the association rule $\{A, B\} \rightarrow \{D\}$ is 100%.
Answer: {A,B}, {A,B,C}, {A,B,E}, {A,B,C,E}.

(d) All transactions that contain {A,B} also contain $\{A, D\}$.
Answer: {A,B}, {A,B,C}, {A,B,E}, {A,B,C,E}.

17. Consider the following set of candidate 3-itemsets:

$$\{p, q, r\}, \{p, q, s\}, \{p, r, s\}, \{p, r, t\}, \{p, r, u\}, \{p, s, u\}, \{p, t, u\},$$
$$\{q, r, s\}, \{q, r, t\}, \{q, s, t\}, \{q, s, u\}, \{q, t, u\}, \{r, s, t\}, \{s, t, u\}$$

(a) Construct a hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items p, s are hashed to the left child of a node, items q, t are hashed to the middle child, while items r, u are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf node irrespective of the number of itemsets already stored at the node.

Condition 2: : If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemsets currently stored at the leaf node is less or equal to $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values. Label the leaf nodes (from left to right) as L_1, L_2, L_3, and so on.

Answer: As shown in Figure 5.8.

(b) Consider a transaction that contains items {p,r,s,t}. List all the leaf nodes in the hash tree to which the transaction will be hashed into.
Answer: The transaction will be hased into L1, L4, L5 and L10.

(c) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.
Answer: $\{p, q, r, s\}$, $\{p, r, s, t\}$, $\{p, r, s, u\}$, $\{p, r, t, u\}$,$\{q, r, s, t\}$ and $\{q, s, t, u\}$.

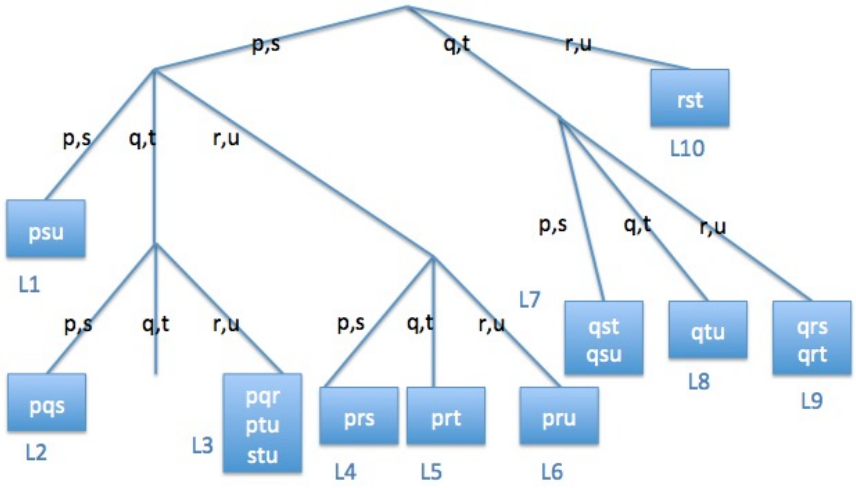


Figure 5.8. Hash Tree

(d) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.
Answer: $\{p, q, r, s\}$, $\{q, r, s, t\}$ and $\{q, s, t, u\}$.

(e) If all the candidate 4-itemsets in part (d) are frequent, is it possible to generate a candidate 5-itemset? If yes, what is the candidate 5-itemset?
Answer: No, since there is no common prefix of length-3 among the 4-itemsets.

18. Consider a medical database that contains N "transactions" (patient records) and d "items". Suppose we are interested to extract rules of the form $X \rightarrow y$, where X is a k-itemset ($1 \leq k \leq d-1$) and y is an item (i.e., the right-hand side of the rule contains only a single item). These are known as discriminative rules. For medical diagnosis, for example, such rules can be used to identify segments of the population who are vulnerable to certain diseases. An example of such rule might be {male, age $>$ 60, alcohol-abuse} $\rightarrow$ {prostate-cancer}.

(a) Count the maximum number of discriminative rules that can be extracted from the database (state your answer in terms of d).
Answer: $d \times (2^{d-1} - 1)$.

(b) Count the maximum number of discriminative rules that can extracted from a given frequent k-itemset X. For example, if $\{p, q, r, s, t\}$

is a frequent 5-itemset, then $\{p, q, r, s\} \rightarrow \{t\}$ is one possible discriminative rule. Note that each extracted rule must include all the items from the k-itemset (either on the left- or right-hand side of the rule).
Answer: k.

(c) Suppose we are interested to extract all the discriminative rules whose support $\geq$ minsup and confidence $\geq$ mincof thresholds. Let X be a frequent k-itemset and X_{-1},X_{-2}, $\cdots$, and X_{-k} be the corresponding set of frequent $(k-1)$-itemsets obtained after removing one of the items from X. What should be the minimum support of the $(k-1)$-itemsets in order to guarantee that the itemset X will generate at least one discriminative rule with confidence $\geq$ minconf. Express your answer in terms of s_X (support of X) and $minconf$.
Answer: Suppose X_{-t} is the itemset with minimum support among all itemsets $X_{-i} \subset X, 1 \leq i \leq k$. Let s_X be the support of X and s_{-t} be the support of X_{-t}. If X can generate at least one high confidence discriminative rule, then $\frac{s_X}{s_{-t}} \geq minconf$. Thus, $s_X \leq s_{-t} \leq \frac{s_X}{minconf}$

(d) Do itemsets that generate discriminative rules have anti-monotone property? In other words, if the itemsets {p,q,r}, {p,q,s}, {p,r,s}, and {q,r,s} do not produce a discriminative rule with confidence $\geq$ minconf, can we conclude that {p,q,r,s} can never produce a discriminative rule with confidence $\geq$ minconf?
Answer: No. A counter example is shown in Table 5.2, assuming minconf = 0.5. Consider the itemset $\{p, q, r\}$. Based on the given transactions, we obtain

$$conf(\{p, q\} \rightarrow \{r\}) = conf(\{p, r\} \rightarrow \{q\}) = conf(\{q, r\} \rightarrow \{p\}) = 1/3,$$

which is less than minconf. Thus, the itemset $\{p, q, r\}$ does not generate any high confidence discriminative rule. However, $conf(\{p, q, r\} \rightarrow \{s\}) = 1 >$ minconf. Therefore, the itemset $\{p, q, r, s\}$ can generate a high confidence discriminative rule, which violates the anti-monotone property.

19. Consider a transaction database that contains six items, {p,q,r,s,t,u}.

(a) Suppose the support of {p,q} is the same as the support of {p,q,s}. Which of the following statement(s) is (are) true:

TID	Transaction
1	p,q,r,s
2	p,q
3	p,q
4	p,r
5	p,r
6	q,r
7	q,r

Table 5.2. Example of transactions

i. Support of {p} is the same as support of {p,s}.
Answer: False.

ii. Confidence of the rule {p,q} → {s} is 100%.
Answer: True.

iii. All transactions that contain item p must also contain item s.
Answer: False.

iv. {p,q,t} is not a closed itemset.
Answer: True. This is because the support of {p,q,t} must be the same support as support of {p,q,s,t} since all transactions that contain $\{p, q\}$ must also contain item s.

(b) Suppose the support of {p,q} is the same as the support of {p,s}. Which of the following statement(s) is (are) true:

i. All transactions that contain item q must also contain item s.
Answer: False.

ii. The confidence of the rule {p,q} → {s} is 100%.
Answer: False.

iii. The support of {p,q} is the same as the support of {p,q,s}.
Answer: False.

iv. {p,q,r} is not a closed itemset.
Answer: False.

20. Consider the following set of candidate 3-itemsets:

$$\{p, q, r\}, \{p, q, s\}, \{p, q, t\}, \{p, r, t\}, \{p, s, t\}, \{q, r, s\}, \{q, r, t\},$$
$$\{q, r, u\}, \{q, s, t\}, \{q, s, u\}, \{s, t, u\}.$$

(a) Construct a hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items p, s are hashed to the left

child of a node, items q, t are hashed to the middle child, while items r, u are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf node irrespective of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemsets stored at the leaf node is less than or equal to $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values.

Answer: See Figure 5.9.

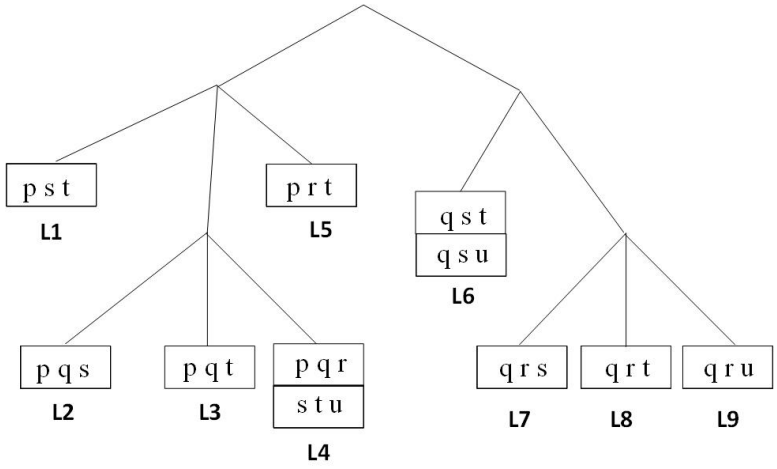


Figure 5.9. Hash Tree

(b) Consider a transaction that contains items {p,q,r,s,u}. Count the number of leaf nodes in the hash tree to which the transaction will be hashed into.

Answer: The transaction will be hashed to 7 leaf nodes (L1, L2, L4, L5, L6, L7, and L9).

(c) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.

Answer:

The candidate 4-itemsets generated are $\{p, q, r, s\}$, $\{p, q, r, t\}$, $\{p, q, s, t\}$, $\{q, r, s, t\}$, $\{q, r, s, u\}$, $\{q, r, t, u\}$, $\{q, s, t, u\}$.

(d) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.

Answer:

The candidate 4-itemsets that survive the pruning step are $\{p, q, r, t\}$ and $\{p, q, s, t\}$.

21. Consider a transaction dataset that contains five items, $\{A, B, C, D, E\}$.

(a) Suppose the support of $\{A, B\}$ is the same as the support of $\{A, B, C\}$, which one of the following statements are true:

i. Support of $\{A\}$ is the same as support of $\{A, C\}$. **False**

ii. The confidence of the rule $\{A, B\} \rightarrow \{C\}$ is 100%. **True**

iii. All transactions that contain item A must also contain item C. **False**

iv. $\{A, B, D\}$ is not a closed itemset. **True** because its support will be identical to the support of $\{A, B, C, D\}$.

(b) Suppose the support of $\{A, B\}$ is the same as the support of $\{A, C\}$, which one of the following statements are true:

i. All transactions that contain item B must contain item C. **False**

ii. The confidence of the rule $\{A, B\} \rightarrow \{C\}$ is 100%. **False**

iii. The support of $\{A, B\}$ is the same as the support of $\{A, B, C\}$. **False**

iv. $\{A, B, D\}$ is not a closed itemset. **False**

(c) Suppose all the transactions that contain $\{A, B\}$ also contain $\{B, C\}$, which one of the following statements are true:

i. The confidence of the rule $\{B, C\} \rightarrow \{A\}$ is 100%. **False**

ii. The support of $\{A\}$ is the same as the support of $\{C\}$. **False**

iii. $\{A, B, D\}$ is not a closed itemset. **True** because its support will be identical to support of $\{A, B, C, D\}$.

(d) Suppose the confidence of the rules $\{A, B\} \rightarrow C$ and $\{A, B\} \rightarrow D$ are identical, which one of the following statements are true:

i. The confidence of the $\{A, B\} \rightarrow \{C, D\}$ is the same as the confidence of $\{A, B\} \rightarrow \{C\}$. **False**

ii. All transactions that contain {A,B,C} also contain {A,B,D}. **False**

iii. $\{A, B\}$ is not a closed itemset. **False**

(e) Consider the lattice structure shown in Figure 5.10.

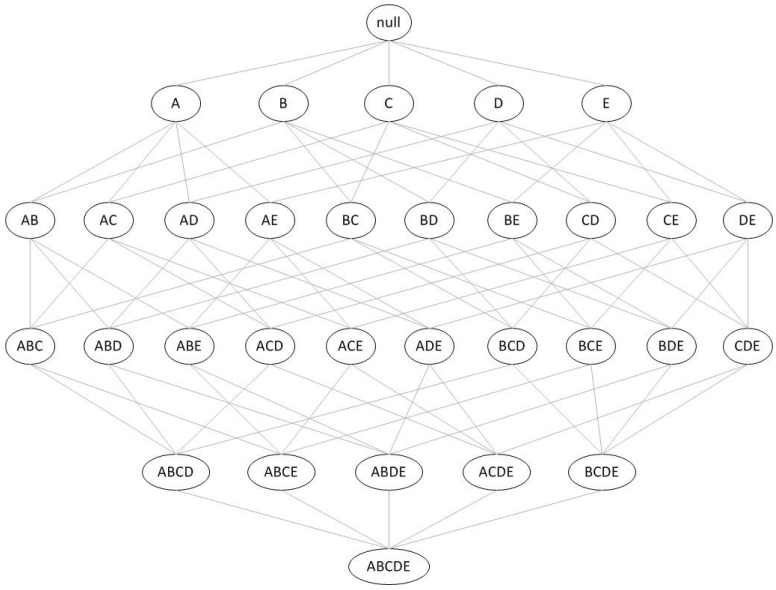


Figure 5.10. Search space for transaction database that contains 6 items.

For each of the following scenarios, list all the itemsets that are guaranteed to be not closed when:

i. Support of $\{A, B\}$ is equal to support of $\{A, B, C\}$.
Answer: $\{A, B\}$, $\{A, B, D\}$, $\{A, B, E\}$, and $\{A, B, D, E\}$.

ii. All the transactions that contain $\{A, B\}$ is a subset of the transactions that contain $\{C\}$.
Answer: $\{A, B\}$, $\{A, B, D\}$, $\{A, B, E\}$, and $\{A, B, D, E\}$.

6

Sequential Patterns and Frequent Subgraphs

6.1 Sequential Pattern Mining

1. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{a, b\}\ \{c\}\ \{d\}\ \{a\} >,$$

assuming there are no timing constraints.

Answer:

$< \{a, b\}\ \{c\} >, < \{a, b\}\ \{d\} >, < \{a, b\}\ \{a\} >, < \{a\}\ \{c\}\ \{d\} >,$
$< \{b\}\ \{c\}\ \{d\} >, < \{a\}\ \{c\}\ \{a\} >, < \{b\}\ \{c\}\ \{d\} >, < \{a\}\ \{d\}\ \{a\} >,$
$< \{b\}\ \{d\}\ \{a\} >, < \{c\}\ \{d\}\ \{a\} >.$

(b) List all the 3-element subsequences contained in the data sequence given in part (a).

Answer:

$< \{a, b\}\ \{c\}\ \{d\} >, < \{a, b\}\ \{c\}\ \{a\} >, < \{a, b\}\ \{d\}\ \{a\} >,$
$< \{a\}\ \{c\}\ \{d\} >, < \{a\}\ \{c\}\ \{a\} >, < \{a\}\ \{d\}\ \{a\} >,$
$< \{b\}\ \{c\}\ \{d\} >, < \{b\}\ \{c\}\ \{a\} >, < \{b\}\ \{d\}\ \{a\} >, < \{c\}\ \{d\}\ \{a\} >.$

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:

$< \{a, b, c\} >,\ \ < \{a, b\}\{c\} >,\ \ < \{a, b\}\{d\} >, < \{a, c\}\{d\} >,$

$< \{a,c,d\} >$, $< \{a\}\{c\}\{d\} >$, $< \{b,c\}\{d\} >$, $< \{b,c\}\{d\} >$,
$< \{b\}\{c\}\{d\} >$, $< \{b\}\{c,d\} >$, $< \{c,d\}\{a\} >$.

Answer:

$< \{a,b,c\}\{d\} >$, $< \{a,b\}\{c\}\{d\} >$, $< \{a,b\}\{c,d\} >$, $< \{a,c,d\}\{a\} >$
, $< \{b\}\{c,d\}\{a\} >$.

(d) Based on your answer in part (b), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.

Answer: $< \{a,b,c\}\{d\} >$, $< \{a,b\}\{c\}\{d\} >$.

2. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{p,q\}\ \{r\}\ \{p,q\} >,$$

assuming no timing constraints.

Answer:

$< \{p,q\},\{r\} >$, $< \{p,q\},\{p\} >$, $< \{p,q\},\{q\} >$, $< \{p\},\{r\},\{p\} >$
, $< \{p\},\{r\},\{q\} >$, $< \{q\},\{r\},\{p\} >$, $< \{q\},\{r\},\{q\} >$, $< \{p\},\{p,q\} >$
, $< \{q\},\{p,q\} >$, $< \{r\},\{p,q\} >$

(b) List all the 3-element subsequences contained in the data sequence given in part (a).

Answer: $< \{p\},\{r\},\{p\} >$, $< \{p\},\{r\},\{q\} >$, $< \{q\},\{r\},\{p\} >$, $<$
$\{q\},\{r\},\{q\} >$

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:

$< \{p,q,r\} >$, $< \{p,q\}\{s\} >$, $< \{p\}\{s,p\} >$, $< \{p\}\{p,q\} >$,
$< \{p,r,s\} >$, $< \{r\}\{s\}\{s\} >$, $< \{q,r\}\{s\} >$, $< \{p,r\}\{s\} >$,
$< \{q\}\{s\}\{p\} >$, $< \{q\}\{r,s\} >$, $< \{r,s\}\{s\} >$.

Answer:

$< \{p,q,r\},\{s\} >$, $< \{p,q\},\{s\},\{p\} >$, $< \{p,r,s\},\{s\} >$, $< \{q,r\},\{s\},\{s\} >$
, $< \{p,r\},\{s\},\{s\} >$, $< \{q\},\{r,s\},\{s\} >$, $< \{p\},\{p,q,r\} >$, $< \{q\},\{p,q\},\{s\} >$

(d) Based on your answer in part (b), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.

Answer:

$< \{p,q,r\},\{s\} >$

3. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{a,b\}\ \{a\}\ \{a,b\} >,$$

assuming no timing constraints.

Answer:

$< \{a,b\}\ \{a\} >$, $< \{a,b\}\ \{b\} >$, $< \{a\}\ \{a\}\ \{a\} >$, $< \{a\}\ \{a\}\ \{b\} >$, $< \{a\}\ \{a,b\} >$, $< \{b\}\ \{a\}\ \{a\} >$, $< \{b\}\ \{a\}\ \{b\} >$, and $< \{b\}\ \{a,b\} >$

(b) List all the 2-element subsequences contained in the data sequence given in part (a).

Answer:

$< \{a,b\}\ \{a\} >$, $< \{a,b\}\ \{a,b\} >$, $< \{a\}\ \{a,b\} >$, $< \{a,b\}\ \{b\} >$, $< \{b\}\ \{a\} >$, $< \{b\}\ \{b\} >$, $< \{a\}\ \{a\} >$, $< \{a\}\ \{b\} >$, and $< \{b\}\ \{a,b\} >$.

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:

$< \{a,b,c\} >$, $< \{a,b\}\{c\} >$, $< \{b\}\{a,c\} >$, $< \{b\}\{c,d\} >$,
$< \{b,c,d\} >$, $< \{a,b\}\{d\} >$, $< \{b,c\}\{d\} >$, $< \{a,c\}\{d\} >$,
$< \{a\}\{c\}\{d\} >$, $< \{b\}\{a,d\} >$, $< \{b\}\{c\}\{d\} >$.

Answer:

$< \{a,b,c,d\} >$, $< \{a,b,c\}\{d\} >$, $< \{a,b\}\{c,d\} >$, $< \{a,b\}\{c\}\{d\} >$, and $< \{b\}\{a,c\}\{d\} >$.

(d) Based on your answer in part (b), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.

Answer:

$< \{a,b,c\}\{d\} >$ and $< \{a,b\}\{c\}\{d\} >$

4. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{p,q\}\ \{r\}\ \{q\} >,$$

assuming no timing constraints.

Answer: $< \{p,q\}\ \{r\} >$, $< \{p,q\}\ \{q\} >$, $< \{p\}\ \{r\}\ \{q\} >$, $< \{q\}\ \{r\}\ \{q\} >$.

(b) List all the 3-element subsequences contained in the data sequence given in part (a).
Answer: $< \{p,q\}\ \{r\}\ \{q\} >, < \{p\}\ \{r\}\ \{q\} >, < \{q\}\ \{r\}\ \{q\} >$.

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:
$< \{p,q,r\} >,\ < \{p,q\}\{s\} >,\ < \{p\}\{s,p\} >, < \{p\}\{p,q\} >,$
$< \{p,r,s\} >,\ < \{r\}\{s\}\{s\} >,\ < \{q,r\}\{s\} >, < \{p,r\}\{s\} >,$
$< \{q\}\{s\}\{p\} >,\ < \{q\}\{r,s\} >,\ < \{r,s\}\{s\} >$.
Answer:
$< \{p,q,r\}\ \{s\} >, < \{p,q\}\ \{s\}\ \{p\} >, < \{p\}\ \{p,q,r\} >, < \{p\}\ \{p,q\}\ \{s\} >,$
$< \{p,r,s\}\ \{s\} >, \{q,r\}\ \{s\}\ \{s\} >, \{p,r\}\ \{s\}\ \{s\} >$, and $\{q\}\ \{r,s\}\ \{s\} >$.

(d) Based on your answer in part (b), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.
Answer: $< \{p,q,r\}\ \{s\} >$.

5. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{a,b\}\ \{a,b\}\ \{c\} >,$$

assuming there are no timing constraints.
Answer: $< \{a,b\}\{a\} >, < \{a,b\}\{b\} >, < \{a,b\}\{c\} >, < \{a\}\{a,b\} >,$
$< \{a\}\{a\}\{c\} >, < \{a\}\{b\}\{c\} >, < \{b\}\{a\}\{c\} >, < \{b\}\{b\}\{c\} >,$
$< \{b\}\{a,b\} >$.

(b) List all the 3-element subsequences contained in the data sequence given in part (a).
Answer: $< \{a,b\}\{a,b\}\{c\} >, < \{a,b\}\{a\}\{c\} >, < \{a,b\}\{b\}\{c\} >,$
$< \{a\}\{a,b\}\{c\} >, < \{a\}\{a\}\{c\} >, < \{a\}\{b\}\{c\} >, < \{b\}\{a,b\}\{c\} >,$
$< \{b\}\{a\}\{c\} >, < \{b\}\{b\}\{c\} >$.

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:
$< \{a,b,c\} >,\ < \{a,b\}\{c\} >,\ < \{a,b\}\{d\} >,\ < \{a,c\}\{d\} >,$
$< \{a\}\{c\}\{d\} >,\ < \{b,c\}\{d\} >,\ < \{b,c,d\} >,\ < \{b\}\{a,c\} >,$
$< \{b\}\{a,d\} >,\ < \{b\}\{c,d\} >,\ < \{b\}\{c\}\{d\} >$.
Answer: $< \{a,b,c\}\{d\} >, < \{a,b,c,d\} >, < \{a,b\}\{c,d\} >,$
$< \{a,b\}\{c\}\{d\} >$ and $< \{b\}\{a,c\}\{d\} >$.

(d) Based on your answer in part (b), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.
Answer: $< \{a,b,c\}\{d\} >$ and $< \{a,b\}\{c\}\{d\} >$.

6. Consider a sequence database containing items p, q, r, s, and t.

(a) List all the unique 3-subsequences contained in the following data sequence:

$$< \{p\}\ \{p,q\}\ \{p\}\ \{r\} >,$$

assuming there are no timing constraints.
Answer: $< \{p\}\ \{p,q\} >$, $< \{p\}\ \{p\}\ \{p\} >$, $< \{p\}\ \{p\}\ \{r\} >$, $< \{p\}\ \{q\}\ \{p\} >$, $< \{p\}\ \{q\}\ \{r\} >$, $< \{p,q\}\ \{p\} >$, $< \{p,q\}, \{r\} >$, $< \{q\}\{p\}\{r\} >$.

(b) List all the unique 4-element subsequences contained in the data sequence given in part (a).
Answer: $< \{p\}\ \{p\}\ \{p\}\ \{r\} >$, $< \{p\}\ \{q\}\ \{p\}\ \{r\} >$, $< \{p\}\ \{p,q\}\ \{p\}\ \{r\} >$

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:
$< \{p,q,r\} >$, $< \{p,q\}\{s\} >$, $< \{p,q\}\{t\} >$, $< \{p\}\{p,q\} >$, $< \{p,r\}\{s\} >$, $< \{p,r\}\{t\} >$, $< \{p\}\{q,r\} >$, $< \{p\}\{s,t\} >$, $< \{q,r\}\{s\} >$, $< \{q,r\}\{t\} >$, $< \{q\}\{s,t\} >$, $< \{r\}\{s,t\} >$.

Answer: $< \{p,q,r\}\ \{s\} >$, $< \{p,q,r\}\ \{t\} >$, $< \{p,q\}\ \{s,t\} >$, $< \{p\}\ \{p,q,r\} >$, $< \{p\}\{p,q\}\{s\} >$, $< \{p\}\{p,q\}\{t\} >$, $< \{p,r\}\ \{s,t\} >$, $< \{p\}\ \{q,r\}\ \{s\} >$, $< \{p\}\ \{q,r\}\ \{t\} >$, $< \{q,r\}\ \{s,t\} >$.

(d) Based on your answer in part (c), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.
Answer: $< \{p,q,r\}, \{s\} >$, $< \{p,q,r\}, \{t\} >$, $< \{p,q\}, \{s,t\} >$, $< \{p,r\}, \{s,t\} >$, $< \{q,r\}, \{s,t\} >$.

7. (a) List all the 3-subsequences contained in the following data sequence:

$$< \{p,q\}\ \{r\}\ \{p,q\} >,$$

assuming no timing constraints.
Answer:
$< \{p,q\}, \{r\} >$, $< \{p,q\}, \{p\} >$, $< \{p,q\}, \{q\} >$, $< \{p\}, \{r\}, \{p\} >$, $< \{p\}, \{r\}, \{q\} >$, $< \{q\}, \{r\}, \{p\} >$, $< \{q\}, \{r\}, \{q\} >$, $< \{p\}, \{p,q\} >$ $< \{q\}, \{p,q\} >$, $< \{r\}, \{p,q\} >$

(b) List all the 3-element subsequences contained in the data sequence given in part (a).

Answer:

$< \{p\}, \{r\}, \{p\} >, < \{p\}, \{r\}, \{q\} >, < \{q\}, \{r\}, \{p\} >, < \{q\}, \{r\}, \{q\} >$
$, < \{p, q\}\{r\}\{p, q\} >, < \{q\}\{r\}, \{p, q\} >, < \{p\}\{r\}\{p, q\} >, < \{p, q\}\{r\}\{q\} >$
$, < \{p, q\}\{r\}\{p\} >$

(c) List all the candidate 4-sequences produced by the candidate generation step of the GSP algorithm from the following frequent 3-sequences:

$< \{p, q, r\} >, \ < \{p, q\}\{s\} >, \ < \{p\}\{s, p\} >, < \{p\}\{p, q\} >,$
$< \{p, r, s\} >, \ < \{r\}\{s\}\{s\} >, \ < \{q, r\}\{s\} >, < \{p, r\}\{s\} >,$
$< \{q\}\{s\}\{p\} >, \ < \{q, r\}\{s\} >, \ < \{q\}\{r, s\} >, \ < \{r, s\}\{s\} > .$

Answer:

$< \{p, q, r\}, \{s\} >, < \{p, q\}, \{s\}, \{p\} >, < \{p, r, s\}, \{s\} >, < \{q, r\}, \{s\}, \{s\} >$
$, < \{p, r\}, \{s\}, \{s\} >, < \{q\}, \{r, s\}, \{s\} >, < \{p\}, \{p, q, r\} >, < \{p\}, \{p, q\}, \{s\} >$

(d) Based on your answer in part (c), list the candidate 4-sequences that survived the candidate pruning step of the GSP algorithm.

Answer:

$< \{p, q, r\}, \{s\} >$

8. Consider the following data sequence s:

Timestamp	Element
10	{p, q, r, s}
20	{p, q, t}
40	{q, r}
50	{q, s, t}
80	{r, s, t}

where the timestamp indicates the time in which events associated with the given element were observed.

(a) State whether each following sequential pattern w is a contiguous subsequence of the data sequence s:

- $w =< \{p\}\{q\}\{r\}\{s\}\{t\} >$
  Answer: Yes
- $w =< \{p\}\{p\}\{q\}\{q\} >$
  Answer: Yes

- $w =< \{p\}\{s\}\{t\} >$
  Answer: No
- $w =< \{p, r\}\{q, r\}\{q, s\} >$
  Answer: No

(b) State whether the following sequential pattern is contained in (i.e., supported by) the data sequence above. Use the following time constraints:

mingap = 0, maxgap = 35, window size = 15, maxspan = 65.

- $w =< \{p, q, r, s, t\} >$
  Answer: Yes
- $w =< \{q, r, s, t\}\{q, r\}\{q, s\} >$
  Answer: Yes
- $w =< \{q, r, s\}\{r, s\}\{r, s\} >$
  Answer: No
- $w =< \{p, q, r\}\{q, r, s\} >$
  Answer: No

9. Consider the following data sequence s:

Timestamp	Element
10	{a, b, c}
20	{a, b, d}
40	{c, d}
50	{a, b, d}
80	{b, e}

where the timestamp indicates the time in which events associated with the given element were observed.

(a) State whether each following sequential pattern w is a contiguous subsequence of the data sequence s:

- $w =< \{a\}\{b\}\{c\}\{d\}\{e\} >$
  Answer: Yes
- $w =< \{a, b, c\}\{a, b\}\{a\} >$
  Answer: No
- $w =< \{a\}\{a\}\{a\} >$
  Answer: No
- $w =< \{b\}\{c\}\{d\} >$
  Answer: Yes

(b) State whether the following sequential pattern is supported by the data sequence s above given the following time constraints:
mingap = 0, maxgap = 35, window size = 15, maxspan = 65.

- $w =< \{a, b, c, d, e\} >$
  Answer: No; violate window size constraint.
- $w =< \{a, b, c, d\}\{e\} >$
  Answer: No; violate maxgap constraint.
- $w =< \{a, b, c, d\}\{a, b, d, e\} >$
  Answer: No; violate maxgap constraint.
- $w =< \{b\}\{c\}\{d\}\{e\} >$
  Answer: Yes.

6.2 Frequent Subgraph Mining

1. Draw all candidate subgraphs obtained from joining the pair of graphs shown in Figure 6.1. Assume the edge-growing method is used to expand the subgraphs.

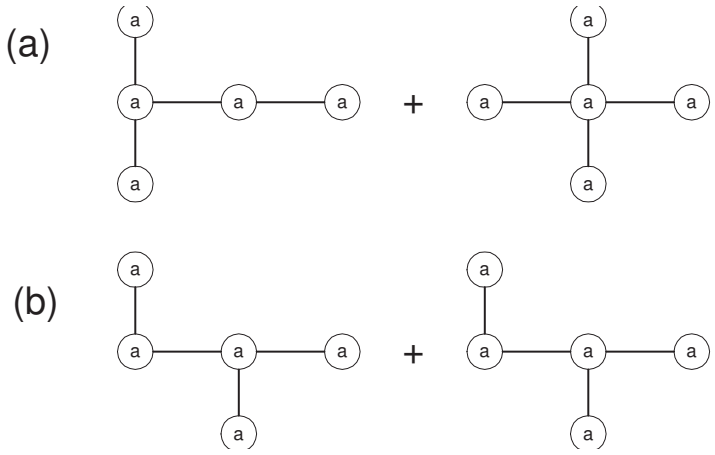


Figure 6.1. Edge growing of subgraphs.

Answer: The candidate subgraphs are shown in Figure 6.2

2. Consider all the frequent 3-subgraphs shown in Figure 6.3.

   (a) Draw all the candidate 4-subgraphs obtained by merging the frequent 3-subgraphs in Figure 6.3. Assume the edge-growing method is used to expand the subgraphs.

   Answer:

   Let F1, F2, and F3 be the three frequent subgraphs shown in Figure 6.3. The candidate subgraphs in Figure 6.4 are obtained as follows:

   i. Merging F1 with F1: G1
   ii. Merging F2 with F2: G2
   iii. Merging F1 with F2: G3
   iv. Merging F3 with F3: G4, G5, and G6
   v. Merging F1 with F3: G7 and G8
   vi. Merging F2 with F3: G9

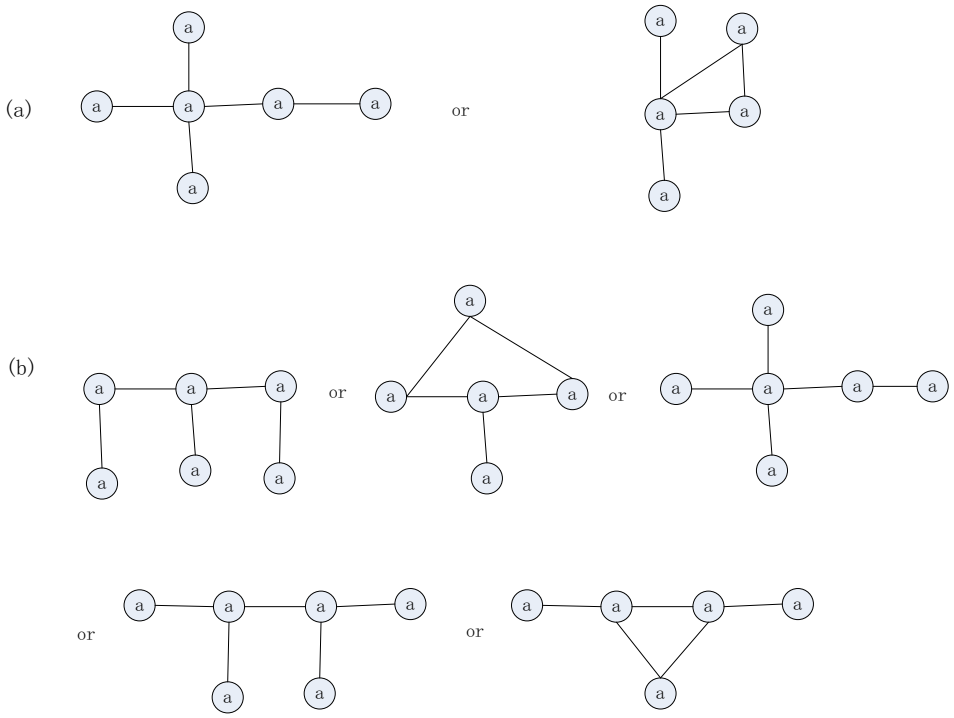


Figure 6.2. Edge growing of subgraphs.

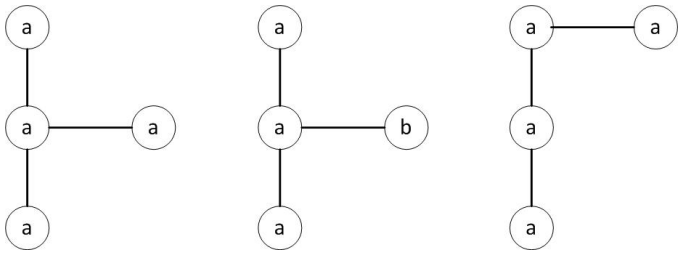


Figure 6.3. Edge growing of subgraphs.

(b) Which of the candidate 4-subgraphs survived the candidate pruning step?

Answer The candidate subgraphs that survive the pruning step are G1, G3, G4, G6, and G7.

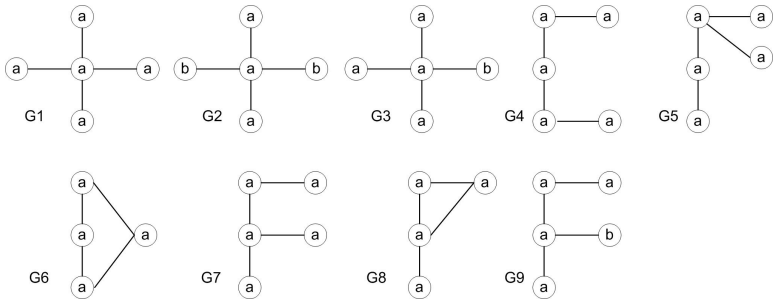


Figure 6.4. Candidate 4-subgraphs.

3. Draw all candidate subgraphs obtained from joining the pair of graphs shown in Figure 6.5. Assume the edge-growing method is used to expand the subgraphs. Make sure you remove all the isomorphic graphs.

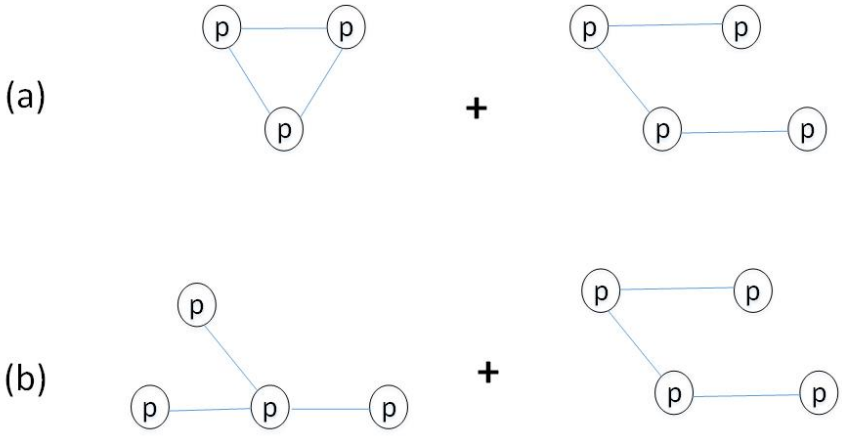


Figure 6.5. Candidate Generation for Subgraphs

Answer: See Figure 6.6.

4. Draw all the candidate 5-subgraphs obtained by merging the frequent 4-subgraphs shown in Figure 6.7 using the FSG algorithm. Assume the edge-growing method is used to expand the subgraphs. Make sure you remove all the isomorphic subgraphs (i.e., subgraphs that are structurally equivalent). You must also specify which pair of frequent 4-subgraphs,

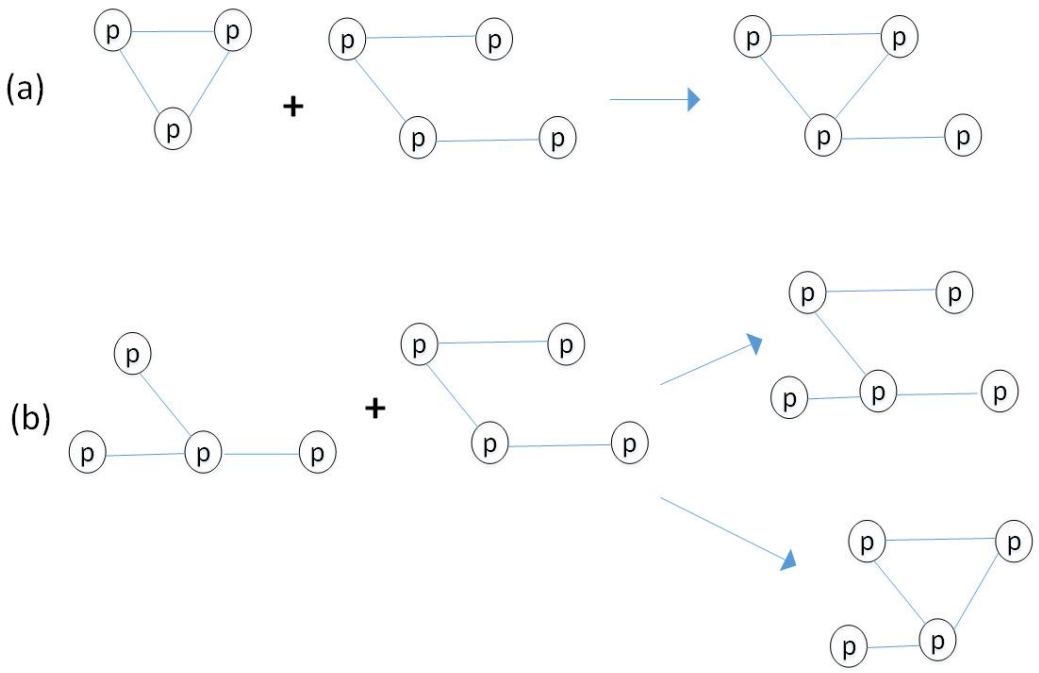


Figure 6.6. Candidate Generation for Subgraphs

(G1, G2), (G1, G3), or (G2,G3), is merged to generate each of your candidate subgraphs. If a candidate can be generated by more than one pair, you only need to list one of such pairs.

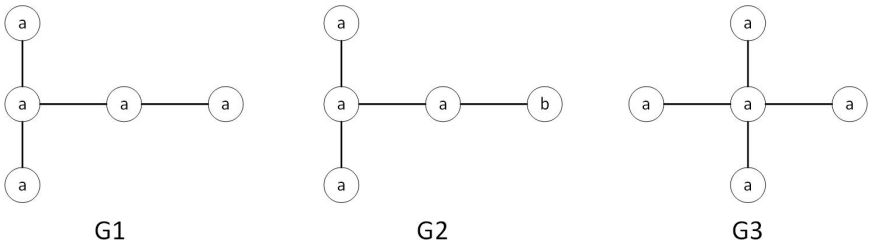


Figure 6.7. Candidate Generation for Subgraphs

Answer:
The solution is shown in Figure 6.8. For each candidate subgraph, the core used to generate the subgraph is shown as shaded nodes in the diagram. Furthermore, some of the subgraphs generated are isomorphic to each other. For example, the subgraphs g1a and g1b are isomorphic, so do (g3a,g3b), (g6a,g6b) and (g8a,g8b). There are altogether 14 distinct candidate 5-subgraphs generated.

Figure 6.8. Candidate subgraphs generated.

ent Subgraphs

5. Draw all the candidate 4-subgraphs obtained by merging the frequent 3-subgraphs shown in Figure 6.9 using the FSG algorithm. Assume the edge-growing method is used to expand the subgraphs. Make sure you remove all the isomorphic subgraphs (i.e., subgraphs that are structurally equivalent). You must also specify which pair of frequent 3-subgraphs, (G1, G2), (G1, G3), or (G2,G3), is merged to generate each of your candidate subgraphs. If a candidate can be generated by more than one pair, you only need to list one of such pairs. Your candidate 4-subgraphs should also include self-joins, i.e., merging each graph with itself.

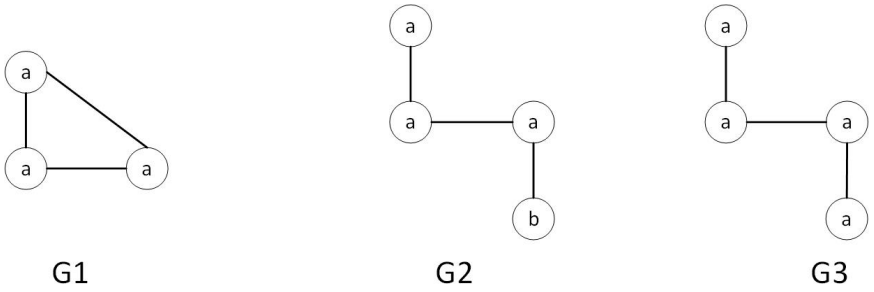


Figure 6.9. Candidate Generation for Subgraphs

Answer: As shown in Figure 6.10.

6. For each question below, draw all the candidate 5-subgraphs generated from joining a pair of frequent 4-subgraphs shown in Figure 6.11 using the method described in the lecture. Assume the edge-growing method is used to expand the subgraphs. Note that we focus on connected subgraphs only with no self-loops and no multiple edges (i.e., there cannot be more than one edge incident on the same pair of vertices). Indicate what are the cores (i.e., the common frequent 3-subgraphs) between the two frequent subgraphs that were joined. For example, the vertices that are part of the core can be shaded (colored) while those that are not part of the core can be left unshaded (no color). Answer no candidates if the graphs to be joined have no common core.

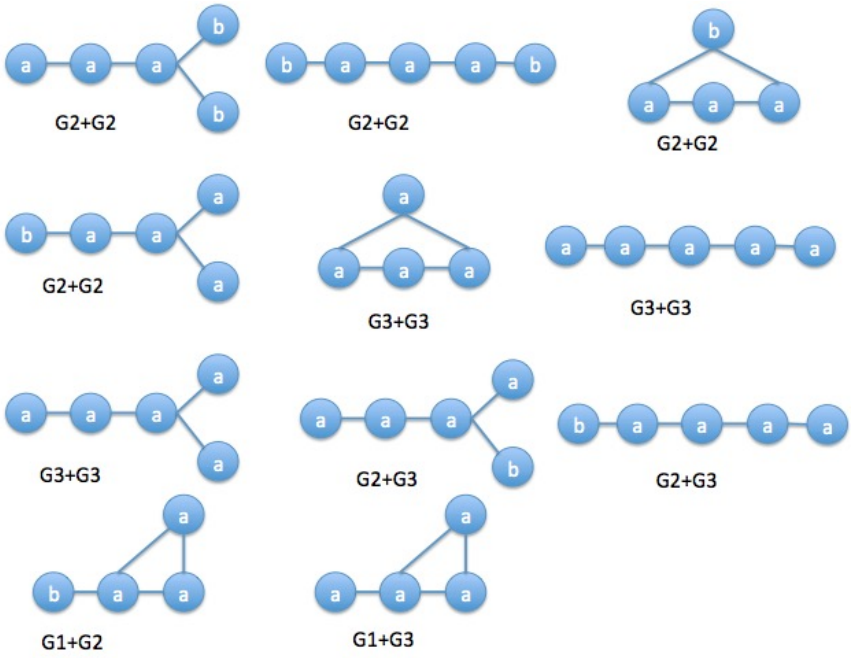


Figure 6.10. Candidate 4-subgraphs by FSG algorithm

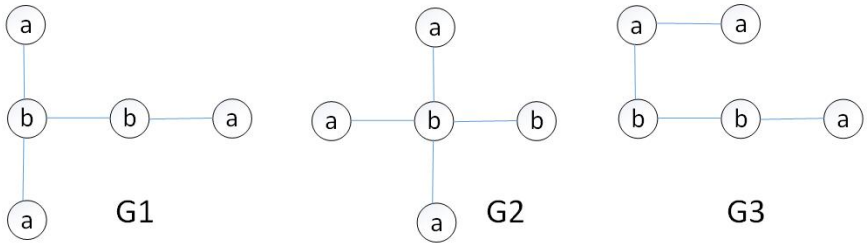


Figure 6.11. Frequent 4-subgraphs.

(a) Draw all the candidate subgraphs after joining G1 with itself.
Answer: See Figure 6.12 (the core is represented by the shaded nodes).

(b) Draw all the candidate subgraphs after joining G2 with itself.
Answer: See Figure 6.13 (the core is represented by the shaded nodes).

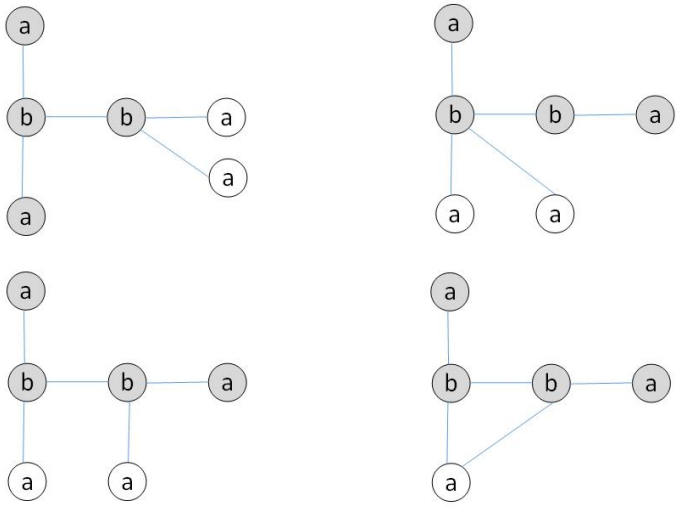


Figure 6.12. Candidate 5-subgraphs.

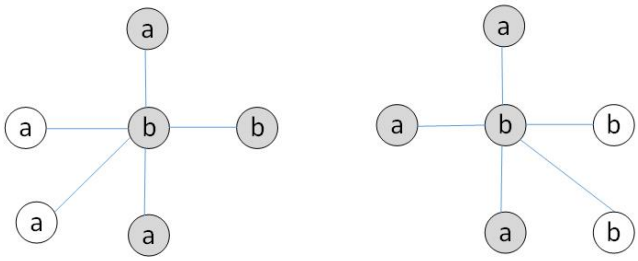


Figure 6.13. Candidate 5-subgraphs.

(c) Draw all the candidate subgraphs after joining G3 with itself.

Answer: See Figure 6.14 (the core is represented by the shaded nodes).

(d) Draw all the candidate subgraphs after joining G1 with G2.

Answer: See Figure 6.15 (the core is represented by the shaded nodes).

(e) Draw all the candidate subgraphs after joining G1 with G3.

Answer: See Figure 6.16 (the core is represented by the shaded nodes).

(f) Draw all the candidate subgraphs after joining G2 with G3.

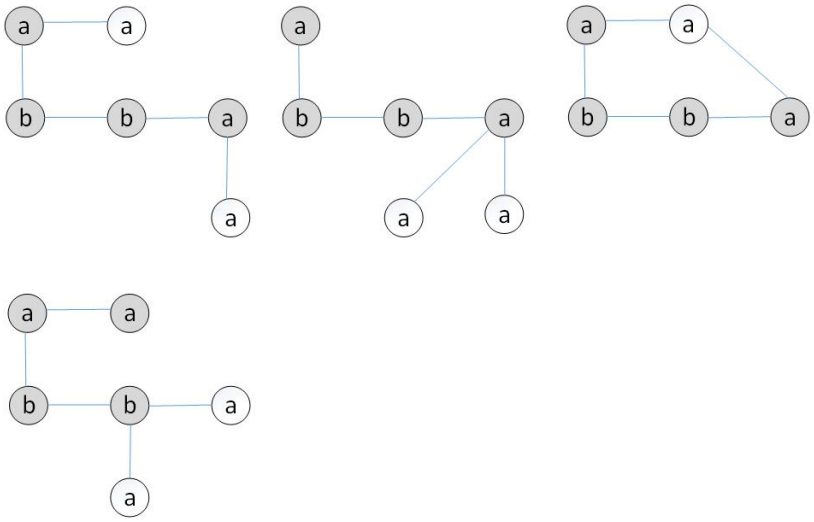


Figure 6.14. Candidate 5-subgraphs.

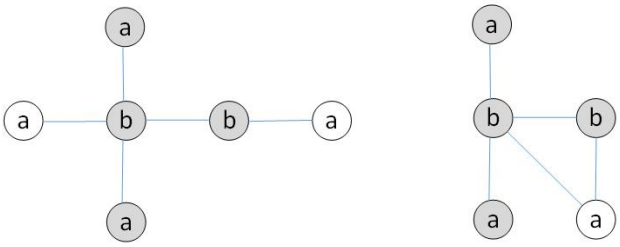


Figure 6.15. Candidate 5-subgraphs.

Answer: No candidates because the graphs do not share a common core.

7. Consider the frequent 3-subgraphs shown in Figure 6.17. Derive the candidate 4-subgraphs obtained by merging the frequent 3-subgraphs using the FSG algorithm. Assume the edge-growing method is used to expand the subgraphs. Make sure you indicate clearly the vertices and edges that are part of the "core" subgraph used for merging. You can do this by shading the core vertices and using thicker lines for the core edges.

   (a) Draw the candidate 4-subgraphs obtained by merging G1 with G2.

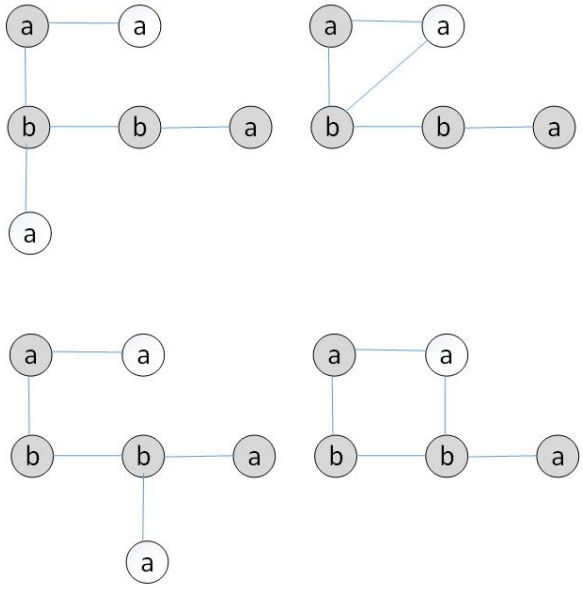


Figure 6.16. Candidate 5-subgraphs.

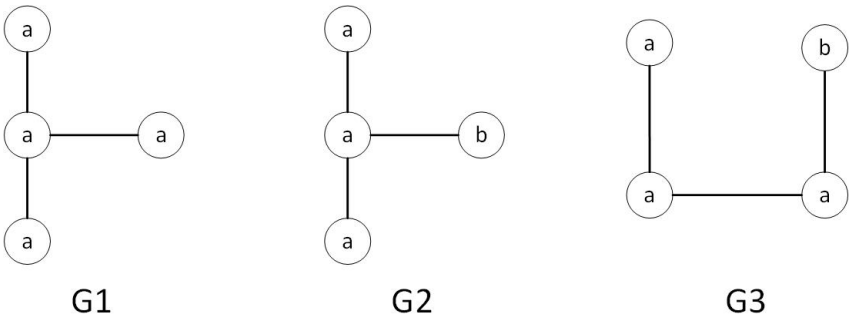


Figure 6.17. Candidate Generation for subgraphs

Answer: See Figure 6.18.

(b) Draw the candidate 4-subgraphs obtained by merging G2 with G3.
Answer: See Figure 6.19.

(c) Draw the candidate 4-subgraphs obtained by merging G1 with G3.
Answer: See Figure 6.20.

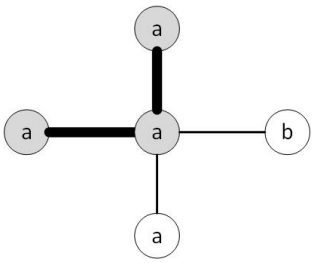


Figure 6.18. Candidates subgraphs obtained by merging G1 with G2.

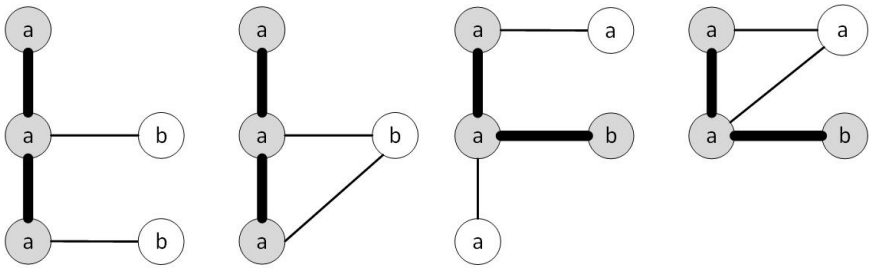


Figure 6.19. Candidate subgraphs obtained by merging G2 with G3

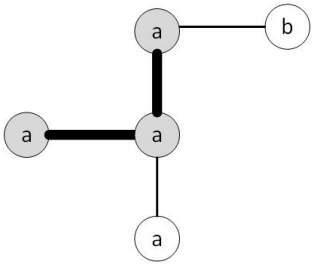


Figure 6.20. Candidate subgraphs obtained by merging G1 with G3

(d) Based on your answers in parts (a), (b), and (c), draw all the candidate 4-subgraphs that survive the candidate pruning step of FSG algorithm.

Answer: All the candidates generated in part (b) by merging G2 with G3 are pruned. The surviving candidate subgraphs are shown in Figure 6.21.

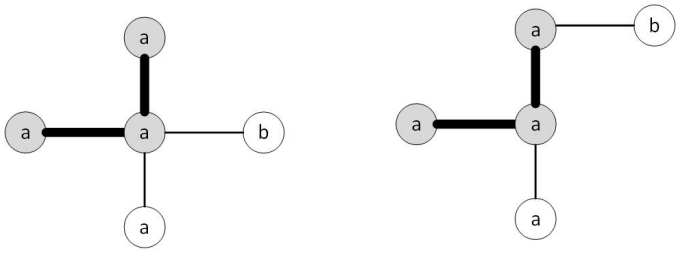


Figure 6.21. Candidate subgraphs generated.

7

Cluster Analysis

7.1 K-means

1. For this question, you need to show that sum-of-squared errors (SSE) is non-increasing when the number of clusters increases. Consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ that contains N observations. Each observation $\mathbf{x}_i$ is a p-dimensional vector of continuous-valued attributes.

   (a) Suppose all the N observations are grouped into a single cluster. Let μ be the centroid of the cluster. Express the total sum-of-squared errors, SSE_T, in terms of $\mathbf{x}_i$, μ and N.
   Answer:

$$\text{SSE}_T = \sum_{i=1}^{N} \|\mathbf{x}_i - \mu\|^2 = \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \mu_j)^2$$

   (b) Show that SSE_T can be decomposed into p separate terms, one for each attribute, i.e., $\text{SSE}_T = \sum_{i=1}^{p} \text{SSE}_i$.
   Answer:

$$\text{SSE}_T = \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \mu_j)^2 = \sum_{j=1}^{p} \sum_{i=1}^{N} (x_{ij} - \mu_j)^2 = \sum_{j=1}^{p} \text{SSE}_j$$

   where SSE_j is the total SSE for dimension j. Therefore, to minimize the total SSE, we can minimize the SSE for each dimension independently.

(c) Based on the results in part (b), it is sufficient to show that SSE is non-increasing with increasing number of clusters assuming an observation $\mathbf{x}_i$ has one-dimensional attribute. Now, suppose all the N observations are grouped into two clusters, C_1 and C_2. Let μ_1 and μ_2 be their corresponding cluster centroids while n_1 and n_2 are their respective cluster sizes. Express the sum-of-squared errors for each cluster, $\text{SSE}^{(i)}$ ($i = 1$ or 2), in terms of $\mathbf{x}_i$, n_i, and μ_i. You need to expand the quadratic term, $(a-b)^2 = a^2 - 2ab + b^2$, and simplify the expression.

Answer: For a given cluster C_j:

$$\begin{aligned}
\text{SSE}^{(j)} &= \sum_{x_i \in C_j} (x_i - \mu_j)^2 = \sum_{x_i \in C_j} (x_i^2 - 2x_i\mu_j + \mu_j^2) \\
&= \sum_{x_i \in C_j} x_i^2 - 2n_j\mu_j^2 + n_j\mu_j^2 \quad (\text{since} \sum_{x_i \in C_j} x_i = \mu_j n_j) \\
&= \sum_{x_i \in C_j} x_i^2 - n_j\mu_j^2
\end{aligned}$$

(d) Rewrite your expression for SSE_T in part (a) in terms of x_i, μ_1, μ_2, n_1, n_2 and N.

Answer: Since $\mu_j = \sum_{x_i \in C_j} x_i/n_j$ and $n_1 + n_2 = N$, therefore

$$\mu = \frac{\sum_i x_i}{N} = \frac{\sum_{x_i \in C_1} x_i + \sum_{x_i \in C_2} x_i}{N} = \frac{n_1}{N}\mu_1 + \frac{n_2}{N}\mu_2$$

Replace the expression into SSE_T:

$$\begin{aligned}
\text{SSE}_T &= \sum_i (x_i - \mu)^2 = \sum_i (x_i^2 - 2x_i\mu + \mu^2) = \sum_i x_i^2 - N\mu^2 \\
&= \sum_i x_i^2 - N\left(\frac{n_1}{N}\mu_1 + \frac{n_2}{N}\mu_2\right)^2 \\
&= \sum_i x_i^2 - \frac{n_1^2}{N}\mu_1^2 - \frac{n_2^2}{N}\mu_2^2 - \frac{2n_1n_2}{N}\mu_1\mu_2 \qquad (7.1)
\end{aligned}$$

(e) Based on your answers in parts (c) and (d), show that $\text{SSE}_T \geq \text{SSE}^{(1)} + \text{SSE}^{(2)}$.

Answer:

$$\begin{aligned} \text{SSE}_1 + \text{SSE}_2 &= \sum_{x_i \in C_1} x_i^2 - n_1\mu_1^2 + \sum_{x_i \in C_2} x_i^2 - n_2\mu_2^2 \\ &= \sum_i x_i^2 - n_1\mu_1^2 - n_2\mu_2^2 \end{aligned} \tag{7.2}$$

Subtracting (7.2) from (7.1)

$$\begin{aligned} \Delta(\text{SSE}) &= \text{SSE}_T - \text{SSE}_1 - \text{SSE}_2 \\ &= n_1\mu_1^2 - \frac{n_1^2}{N}\mu_1^2 + n_2\mu_2^2 - \frac{n_2^2}{N}\mu_2^2 - \frac{2n_1n_2}{N}\mu_1\mu_2 \\ &= n_1\mu_1^2(1 - \frac{n_1}{N}) + n_2\mu_2^2(1 - \frac{n_2}{N}) - \frac{2n_1n_2}{N}\mu_1\mu_2 \\ &= \frac{n_1n_2}{N}\mu_1^2 + \frac{n_1n_2}{N}\mu_2^2 - \frac{2n_1n_2}{N}\mu_1\mu_2 \\ &= \frac{n_1n_2}{N}(\mu_1 - \mu_2)^2 \\ &\geq 0 \end{aligned} \tag{7.3}$$

2. Consider the following set of one-dimensional points: $\{5, 7, 16, 18, 24, 26, 34, 38\}$.

   (a) Suppose we apply k-means clustering to obtain three clusters, A, B, and C. If the three initial centroids are located at (15, 25, 31), respectively, show the clustering results after assigning each point to their closest centroid.

   Answer: {5,7,16,18}, {24,26}, {34,38}

   (b) Based on your answer in part (a), recompute the new locations of the centroids for A, B, and C. Compute also their overall SSE.

   Answer: New centroids are 11.5, 25, 36. SSE is 135.

   (c) What are the locations of the cluster centroids when the algorithm converges? Compute also their overall SSE.

   Answer: Same as part (b).

   (d) Explain how well you think bisecting kmeans will perform on the same data set (compared to regular kmeans). State your reason clearly.

   Answer: Same as part (b).

(e) Suppose we add 199 equally spaced data points between 5 and 7 (i.e., 5.01, 5.02, $\cdots$, 6.99) to the previous data set. How do the clusters look like after applying kmeans with $k = 3$?

Answer: {5,$\cdots$,7}, {16,18,24,26}, {34,38}

3. For each situation shown below, explain whether it is a feasible solution of k-means clustering (for the given value of k). The locations of the centroids are marked as $\times$.

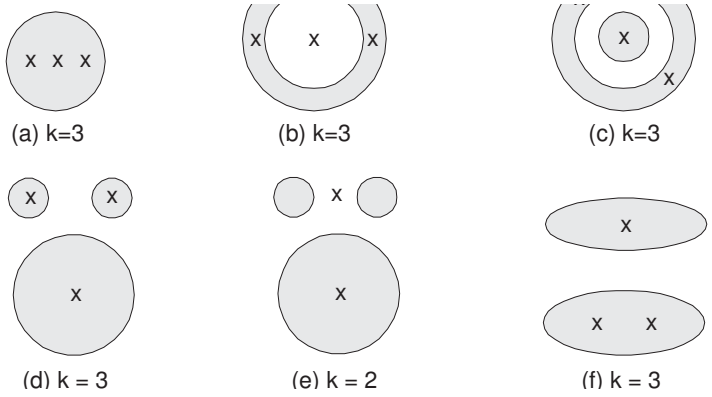


Figure 7.1. K-means clustering.

Answer:

(a) No. The middle centroid is stable; but the two outer centroids are not.

(b) No. The centroids on the ring will be pulled toward the center of the ring.

(c) No. The centroids on the ring will be pulled toward the center of the ring.

(d) No. Points located on the northern border of the larger circle are closer to the centroids of the smaller circle than to the centroid of the larger circle. As a result, the centroids of the two smaller circles will be pulled downward.

(e) No. Argument is the same as part (d).

(f) Yes. This solution should be feasible because it is a stable (though not globally optimal) solution.

4. For each situation shown in Figure 7.2, explain whether it is a feasible solution of k-means clustering (for the given value of k). We consider a solution to be feasible if k-means converges to the centroids shown in the diagram (with the proper choice of initial centroid). The locations of the centroids are marked as $\times$.

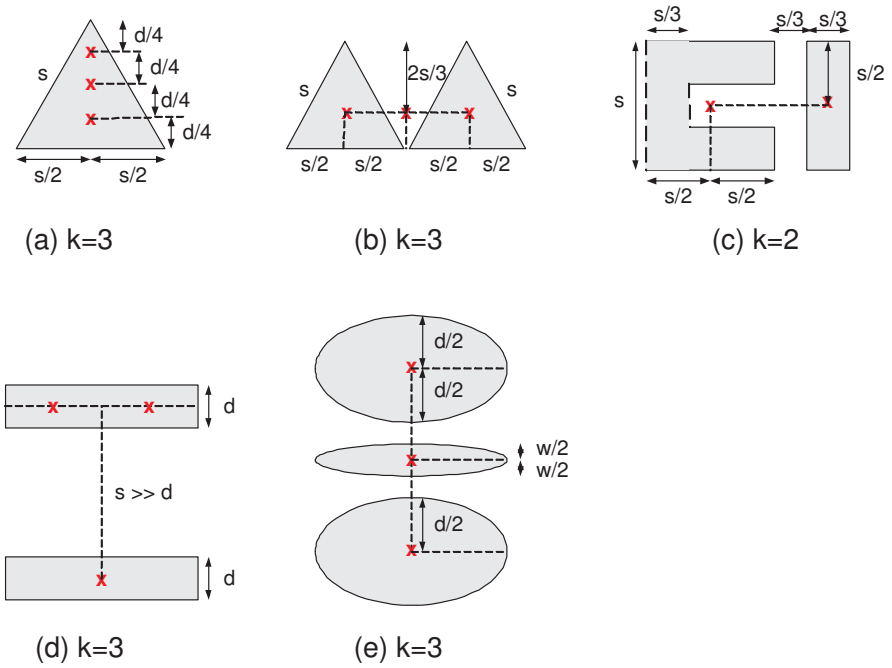


Figure 7.2. K-means clustering.

Answer:

(a) Not feasible.

(b) Not feasible.

(c) Not feasible.

(d) Feasible.

(e) Not feasible.

5. Consider the following set of one-dimensional points: {0.1, 0.2, 0.45, 0.55, 0.8, 0.9}. All the points are located in the range between [0,1].

(a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.4, 1}, respectively, show the cluster assignments and locations of the centroids after the first three iterations by filling out the following table.

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.20	0.45	0.55	0.80	0.90	A	B	C
0	-	-	-	-	-	-	0.00	0.40	1.00
1									
2									
3									

Answer:

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.20	0.45	0.55	0.80	0.90	A	B	C
0	-	-	-	-	-	-	0.00	0.40	1.00
1	A	B	B	B	C	C	0.10	0.40	0.85
2	A	A	B	B	C	C	0.15	0.50	0.85
3	A	A	B	B	C	C	0.15	0.50	0.85

$$SSE = (0.05)^2 \times 6 = 0.0015$$

(b) For the dataset given in part (a), is it possible to obtain empty clusters? If possible, what are the values of the initial centroids? If not, state why.

Answer: Yes. For example, if the centroids are located at 0, 0.90, and 1.0. There won't be any points assigned to cluster C.

(c) Show the clustering results obtained using bisecting kmeans (with k=3). Comparing against the result for k-means, which method is better to cluster this dataset?

Answer:

There are three possible solutions (depending on the choice of initial centroids).

The first solution is when bisecting k-means initially partitions the data into the following two equal sized clusters:

Cluster 1 contains 0.10, 0.20, and 0.45

Cluster 2 contains 0.55, 0.80 and 0.90

Next, it will select one of the two clusters and partition it into two smaller clusters. Since both clusters have the same SSE, it

randomly chooses one of them and split it into two. For example, if cluster 1 is selected for splitting, the results would be:

Cluster 1 contains 0.10 and 0.20

Cluster 2 contains 0.45

Cluster 3 contains 0.55, 0.80, and 0.90.

In this case, the k-means results are better than bisecting k-means on this dataset.

The second and third solutions occur when bisecting k-means initially partitions the data into two unbalanced clusters, either:

Cluster 1 contains 0.10, 0.20, 0.45, and 0.55

Cluster 2 contains 0.80 and 0.90

or

Cluster 1 contains 0.10, 0.20

Cluster 2 contains 0.45, 0.55, 0.80 and 0.90

In both cases, the clusters obtained by bisecting k-means will be identical to the clusters found by regular k-means.

6. K-means does not always converge to the optimal solution as it is sensitive to the choice of initial centroids. The centroids are often randomly initialized to a subset of the data points to be clustered.

   (a) If the data set contains 100 data points, how many times do you need to repeat the k-means algorithm (each time with a different initialization of centroids) to ensure there is a 50% chance an optimal k-means solution would be found? Assume the number of clusters is 10 and that each configuration of initial centroids leads to a distinct clustering solution (i.e., yields a different SSE).

   Answer:

   First, given 100 data points, there are $M = \binom{100}{10}$ possible configurations of the initial centroids. If each configuration yields a distinct solution, in which only one of them gives an optimal solution, the probability you achieve the optimal solution by randomly choosing one of the initial centroid configuration is $1/M$. If you repeat k-means twice by randomly choosing two distinct initial centroids configurations (i.e., sampling without replacement), then the probability one of the configuration yields an optimal solution is $2/M$. Thus, if you had randomly chosen p distinct configurations, the

probability one of the configuration is optimal is p/M. We want $p/M = 0.5$, so the number of times we should repeat k-means is

$$p = 0.5 \times M = 0.5 \times \binom{100}{10} = 8.66 \times 10^{12}$$

(b) If the number of data points increases from 100 to 200 (but number of clusters is still 10), will it improve or diminish your chance of finding the optimal k-means solution by randomly choosing a subset of the data points to be the initial centroids?

Answer:

Since $\binom{200}{10} > \binom{100}{10}$ and the probability that a random initial configuration yields an optimal solution is $1/M$, it will diminish the chance.

(c) If the number of data points is fixed but number of clusters (and number of initial centroids) increases, will it improve or diminish your chance of finding the optimal k-means solution by randomly choosing a subset of the data points to be the initial centroids?

Answer:

Since $\binom{100}{k} > \binom{100}{10}$ only if k is between 10 and 89, it will generally diminish your chance unless the number of clusters is at least 90.

7. Consider the following set of one-dimensional data points: {0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9}.

(a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.25, 0.6}, respectively, show the cluster assignments and locations of the centroids after the first three iterations.

Answer:

	Cluster assignment of data points							Centroid Locations		
Iter	0.10	0.20	0.40	0.50	0.60	0.80	0.90	A	B	C
0	-	-	-	-	-	-	-	0.00	0.25	0.60
1	A	B	B	C	C	C	C	0.1	0.3	0.7
2	A	A	B	B	C	C	C	0.15	0.45	0.77
3	A	A	B	B	B	C	C	0.15	0.5	0.85

(b) Compute the SSE of the k-means solution (after 3 iterations).

Answer:

$$SSE = 0.03$$

(c) Apply bisecting k-means (with k=3) on the data. First, apply k-means on the data with k=2 using initial centroids located at {0.1, 0.9}.

	Cluster assignment of data points							Centroid	
Iter	0.10	0.20	0.40	0.50	0.60	0.80	0.90	A	B
0	-	-	-	-	-	-	-	0.10	0.90
1									
2									

Next, compute the SSE for each cluster (make sure you indicate the SSE values in your answer). Choose the cluster with larger SSE value and split it further into 2 sub-clusters. You can choose the two data points with the smallest and largest values as your initial centroids. For example, if the cluster to be split contains data points (0.20, 0.40, 0.60, and 0.80), then the centroids should be initialized to 0.20 and 0.80. Show the clustering solution produced obtained applying bisecting k-means.

Answer:

	Cluster assignment of data points							Centroid	
Iter	0.10	0.20	0.40	0.50	0.60	0.80	0.90	A	B
0	-	-	-	-	-	-	-	0.10	0.90
1	A	A	A	B	B	B	B	0.50	0.90
2	A	A	A	B	B	C	C	-	-

$$SSE = 0.056$$

(d) Compare the results of k-means clustering against bisecting k-means. Which clustering method is more effective for the given data set?

Answer:

According to SSE, k-meas has smaller SSE value, so k-means is better in this case.

8. Consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where each $\mathbf{x}_i$ is a d-dimensional feature vector. The goal of k-means clustering algorithm is to minimize

the following objective function:

$$\min_{\{\mathbf{c}\},\mathbf{W}} \sum_{i=1}^{N}\sum_{j=1}^{k} W_{ij}\|\mathbf{x}_i - \mathbf{c}_j\|^2, \tag{7.4}$$

where k is the number of clusters, $\mathbf{c}_j$ is the centroid of cluster j, and $\mathbf{W}$ is an $N \times k$ cluster membership matrix ($W_{ij} = 1$ if data point $\mathbf{x}_i$ belongs to cluster j or 0 otherwise). After assigning each data point to its nearest centroid, the centroid location c_k can be updated as follows:

$$\mathbf{c}_k = \frac{\sum_{i=1}^{N} W_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} W_{ik}} \tag{7.5}$$

The denominator term $\sum_{i=1}^{N} W_{ik}$ corresponds to the number of data points assigned to cluster k.

The update formula for cluster centroids depends on the choice of objective function. For each scenario described below, derive the update formula for the cluster centroids.

(a) The original k-means clustering algorithm assumes each point is equally important in determining the clusters. However, in reality, some data points could be outliers or noise. One way to address this would be to assign weights to each point, giving lower weights for points that are considered outliers or noise (assume the outliers and noise points can be pre-determined during preprocessing). Let $0 \leq v_i \leq 1$ be the weight for a data point $\mathbf{x}_i$. Show how the k-means objective function can be updated to accommodate weights on each data points. Next, derive a mathematical expression for the centroid update formula (analogous to Equation 7.5). You may assume the weights of the data points are known when applying the "weighted k-means" algorithm.

Answer: The modified objective function is

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{k} v_i W_{ij}\|\mathbf{x}_i - \mathbf{c}_j\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{k} v_i W_{ij}(\mathbf{x}_i - \mathbf{c}_j)^T(\mathbf{x}_i - \mathbf{c}_j)$$

Taking its partial derivative with respect to $\mathbf{c}_k$ and setting it to zero yields

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{c}_k} &= \sum_{i=1}^{N} v_i W_{ik}\Big[-2\mathbf{x}_i + 2\mathbf{c}_k\Big] = 0 \\ \Longrightarrow \quad \mathbf{c}_k &= \frac{\sum_{i=1}^{N} v_i W_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} v_i W_{ik}}\end{aligned}$$

which is similar to the original k-means approach except the centroids are computed based on a weighted average of the data points associated with the cluster.

(b) The original formulation for k-means assumes a squared Euclidean distance function,

$$d(\mathbf{x}_i, \mathbf{c}_j) = \|\mathbf{x}_i - \mathbf{c}_j\|^2 = \sum_{p=1}^{d}(x_{ip} - c_{jp})^2.$$

Suppose the objective function is modified to 1 - cosine similarity:

$$d(\mathbf{x}_i, \mathbf{c}_j) = 1 - \mathbf{x}_i^T\mathbf{c}_j = 1 - \sum_{p=1}^{d} x_{ip}c_{jp},$$

where each data point $\mathbf{x}_i$ is assumed to have been normalized to unit length, i.e., $\mathbf{x}_i^T\mathbf{x}_i = 1$. Derive a mathematical expression for the centroid update formula (analogous to Equation 7.5) using the new objective function. Note that you should constrain the centroids to have unit length, i.e., $\mathbf{c}_i^T\mathbf{c}_i = 1$.

Answer: The Lagrangian formulation for the constrained optimization problem can be written as follows

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{k} W_{ij}(1 - \mathbf{x}_i^T\mathbf{c}_j) + \sum_{j=1}^{k}\lambda_j(\mathbf{c}_j^T\mathbf{c}_j - 1)$$

Taking its partial derivative with respect to $\mathbf{c}_m$ and setting it to zero yields

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{c}_m} &= -\sum_{i=1}^{N} W_{im}\mathbf{x}_i + 2\lambda_m \mathbf{c}_m = 0 \\
&\Longrightarrow \mathbf{c}_m = \frac{\sum_{i=1}^{N} W_{im}\mathbf{x}_i}{2\lambda_m} \\
&\Longrightarrow \mathbf{c}_m^T \mathbf{c}_m = 1 = \frac{1}{4\lambda_m^2} \sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j \\
&\Longrightarrow \lambda_m = \frac{1}{2}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j} \\
&\Longrightarrow \mathbf{c}_m = \frac{\sum_{i=1}^{N} W_{im}\mathbf{x}_i}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j}}
\end{aligned} \tag{7.6}$$

9. Consider the following set of one-dimensional points: {0.1, 0.25, 0.45, 0.55, 0.8, 0.9}. All the points are located in the range between [0,1].

   (a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.4, 1}, respectively, show the cluster assignments and locations of the updated centroids after the first three iterations by filling out the following table.

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B	C
0	A	B	B	B	C	C	0.00	0.40	1.00
1	A	A	B	B	C	C	0.1	0.42	0.85
2	A	A	B	B	C	C	0.18	0.5	0.85
3	A	A	B	B	C	C	0.18	0.5	0.85

   Calculate the overall sum-of-squared errors of the clustering after the third iteration, where:

$$\text{SSE} = \sum_{x_i \in \text{A}} (x_i - \mu_1)^2 + \sum_{x_i \in \text{B}} (x_i - \mu_2)^2 + \sum_{x_i \in \text{C}} (x_i - \mu_3)^2$$

   and μ_js are the centroids of the 3 clusters, A, B, and C.

Answer: SSE = 0.0213

(b) For the dataset given in part (a), is it possible to obtain empty clusters? If possible, what are the values of the initial centroids? If not, state why.

Answer: Yes. If two of the centroids are randomly initialized to the left of the leftmost points or to the right of the rightmost points. For example, if the initial centroids are (0, 0.05, 0.8), then the first cluster is empty.

(c) Show the clustering results obtained using bisecting kmeans (with k=3). Start by partitioning the data into 2 clusters, with initial centroids 0 and 1. Then take the cluster with larger SSE and partition it into 2 clusters. Compare the results against k-means. Which method is better to cluster the given dataset?

Answer: After the first binary partition:

	Cluster assignment of data points						Centroid Locations	
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B
0	A	A	A	B	B	B	0.0	1.0
1	A	A	A	B	B	B	0.27	0.75
2	A	A	A	B	B	B	0.27	0.75
3	A	A	A	B	B	B	0.27	0.75

The resulting sum-of-squared errors for the clusters are

$$\text{SSE (cluster A)} = 0.0617, \quad \text{SSE (cluster B)} = 0.065$$

Next, we partition cluster B into 2, denoted as B1 and B2, with initial centroids at 0.55 and 0.80.

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B1	B2
0	A	A	A	B1	B2	B2	-	0.55	0.80
1	A	A	A	B1	B2	B2	-	0.55	0.85
2	A	A	A	B1	B2	B2	-	0.55	0.85
3	A	A	A	B1	B2	B2	-	0.55	0.85

The SSE of the bisecting kmeans is 0.0667, which is larger than the SSE for k-means. Therefore, K-means is more effective on this data set. Intuitively, you can see that bisecting K-means always partition the middle cluster into 2 after the first step. So, the middle cluster (which contains the points 0.45 and 0.55) will always be in separate clusters, which is not the optimal solution.

10. Consider the following set of one-dimensional data points: {0.1, 0.2, 0.42, 0.5, 0.6, 0.8, 0.9}.

    (a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.25, 0.6}, respectively, show the cluster assignments and locations of the centroids after the first three iterations by filling out the following table.

    Calculate the overall sum-of-squared errors of the clustering after the third iteration, where:

$$\text{SSE} = \sum_{x_i \in \text{A}} (x_i - \mu_1)^2 + \sum_{x_i \in \text{B}} (x_i - \mu_2)^2 + \sum_{x_i \in \text{C}} (x_i - \mu_3)^2$$

    and μ_js are the centroids of the 3 clusters, A, B, and C.

    Answer:

	Cluster assignment of data points							Centroid Locations		
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B	C
0	-	-	-	-	-	-	-	0.00	0.25	0.60
1	1	2	2	3	3	3	3	0.10	0.31	0.70
2	1	1	2	2	3	3	3	0.15	0.46	0.767
3	1	1	2	2	2	3	3	0.15	0.507	0.85

    Total SSE = 0.0263

    (b) Apply bisecting k-means (with k=3) on the data. First, apply k-means on the data with k=2 using initial centroids located at {0.1, 0.9}.

    Next, compute the SSE for each cluster (make sure you indicate the SSE values in your answer). Choose the cluster with larger SSE value and split it further into 2 sub-clusters. You can choose the pair of points with the smallest and largest values as your initial centroids. For example, if the cluster to be split contains data points (0.20, 0.40, 0.60, and 0.80), then the centroids should be initialized to 0.20 and 0.80. Show the clustering solution produced obtained applying bisecting k-means.

    Answer:

	Cluster assignment of data points							Centroid	
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B
0	-	-	-	-	-	-	-	0.10	0.90
1	1	1	1	1	2	2	2	0.305	0.767
2	1	1	1	1	2	2	2	0.305	0.767

SSE (A) = 0.104, SSE (B) = 0.047, we split cluster A into two, with initial centroids at 0.10 and 0.50.

	Cluster assignment of data points							Centroid Locations		
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B	C
0	-	-	-	-	-	-	-	0.10	0.50	0.767
1	1	1	3	3	2	2	2	0.15	0.46	0.767
2	1	1	3	3	2	2	2	0.15	0.46	0.767

Total SSE = 0.0549

(c) Compare the results of k-means clustering against bisecting k-means. Which clustering method is more effective for the given data set?

Answer: The k-means clustering is more effective because bisecting k-means always split the middle cluster into two parts.

11. Consider the one-dimensional data given in the previous question. Apply fuzzy k-means algorithm with fuzzifier = 2 to obtain three clusters, A, B, and C.

(a) If the initial centroids are located at {0, 0.25, 0.6}, respectively, show the fuzzy cluster assignments and locations of the centroids after the first three iterations by filling out the following table.

If you assign each data point to the cluster with highest fuzzy score, are the resulting clusters similar to the k-means clustering results?

Answer:

	Fuzzy cluster assignment of data points							Centroid Locations		
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B	C
0	-	-	-	-	-	-	-	0.00	0.25	0.60
1: A	0.674	0.058	0.080	0.033	0	0.052	0.084	0.122	0.259	0.666
B	0.299	0.928	0.486	0.133	0	0.111	0.161			
C	0.027	0.014	0.434	0.833	1	0.837	0.755			
2: A	0.980	0.363	0.169	0.115	0.018	0.036	0.074	0.129	0.332	0.713
B	0.018	0.627	0.583	0.285	0.035	0.056	0.109			
C	0.001	0.010	0.249	0.600	0.947	0.908	0.817			
3: A	**0.983**	**0.764**	0.077	0.112	0.047	0.016	0.050	0.143	0.440	0.761
B	0.015	0.221	**0.847**	**0.548**	0.144	0.033	0.093			
C	0.002	0.015	0.076	0.340	**0.809**	**0.951**	**0.857**			

The cluster assignment is shown in bold. The answer is not the same as k-means. However, if you continue the clustering process for another iteration, the solution is given below.

	Fuzzy cluster assignment of data points							Centroid Locations		
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B	C
4: A	**0.980**	**0.939**	0.005	0.027	0.058	0.003	0.030	0.149	0.474	0.817
B	0.016	0.052	**0.992**	**0.924**	**0.472**	0.012	0.081			
C	0.004	0.009	0.003	0.050	0.469	**0.985**	**0.889**			

which is similar to k-means results.

(b) Repeat the previous analysis with k=2 using initial centroids located at {0.1, 0.9}. Summarize the results for the first three iterations by filling out the following table.

Based on their fuzzy scores, which data point has the most "uniform" fuzzy score distribution? Such a data point are expected to be located at the boundary between the two clusters.

Answer:

	Fuzzy cluster assignment of data points							Centroid Locations	
Iter	0.10	0.20	0.42	0.50	0.60	0.80	0.90	A	B
0	-	-	-	-	-	-	-	0.10	0.90
1: A	1.000	0.980	0.692	0.500	0.265	0.020	0.000	0.239	0.758
B	0.000	0.020	0.308	0.500	0.735	0.980	1.000		
2: A	0.957	0.995	0.778	0.495	0.161	0.006	0.044	0.246	0.750
B	0.043	0.005	0.222	0.505	0..839	0.994	0.956		
3: A	0.952	0.993	0.782	**0.491**	0.152	0.008	0.050	0.248	0.748
B	0.048	0.007	0.218	**0.509**	0.848	0.992	0.950		

The data point with the most uniform distribution is located at 0.50.

12. Consider the following set of one-dimensional data points: {0.10, 0.15, 0.23, 0.50, 0.60, 0.84, 0.95}.

(a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0.10, 0.23, 0.84}, respectively, show the cluster assignments and locations of the centroids after the first three iterations.

Calculate the overall sum-of-squared errors of the clustering after the third iteration, where:

$$\text{SSE} = \sum_{x_i \in \text{A}} (x_i - \mu_1)^2 + \sum_{x_i \in \text{B}} (x_i - \mu_2)^2 + \sum_{x_i \in \text{C}} (x_i - \mu_3)^2$$

and μ_js are the centroids of the 3 clusters, A, B, and C.

Answer:

Iter	Cluster assignment of data points							Centroid Locations		
	0.10	0.15	0.23	0.50	0.60	0.84	0.95	A	B	C
0	-	-	-	-	-	-	-	0.10	0.23	0.84
1	A	A	B	B	C	C	C	0.125	0.365	0.797
2	A	A	A	B	C	C	C	0.160	0.500	0.797
3	A	A	A	B	B	C	C	0.160	0.550	0.895

SSE = 0.01965

(b) Apply bisecting k-means (with k=3) on the data. First, apply regular k-means with k=2 using the initial centroids located at {0.10, 0.95}.

Next, compute the SSE for each cluster (make sure you indicate the SSE values in your answer). Choose the cluster with larger SSE value and split it further into 2 sub-clusters by filling out the table below:

Answer:

Results after first split:

Iter	Cluster assignment of data points							Centroid	
	0.10	0.15	0.23	0.50	0.60	0.84	0.95	A	B
0	-	-	-	-	-	-	-	0.10	0.95
1	A	A	A	A	B	B	B	0.245	0.797
2	A	A	A	A	B	B	B	0.245	0.797

Results after second split:

Iter	Cluster assignment of data points							Centroid Locations		
	0.10	0.15	0.23	0.50	0.60	0.84	0.95	A	B	C
0	-	-	-	-	-	-	-	0.100	0.500	0.797
1	A	A	A	C	B	B	B	0.160	0.500	0.797
2	A	A	A	C	B	B	B	0.160	0.500	0.797
3	A	A	A	C	B	B	B	0.160	0.500	0.797

SSE = 0.07267

(c) Compare the overall SSE of k-means clustering against bisecting k-means. Which clustering method is more effective for the given data set?

Answer: K-means has lower SSE compared to bisecting k-means on the given dataset.

13. Consider the one-dimensional data given in the previous question. Apply fuzzy k-means algorithm with fuzzifier = 2 to obtain three clusters, A, B, and C. If the initial centroids are located at {0.10, 0.23, 0.84}, respec-

tively, show the fuzzy cluster assignments and locations of the centroids after the first four iterations.

If you assign each data point to the cluster with highest fuzzy score, are the resulting clusters similar to the k-means clustering results?

Answer:

	Fuzzy score of data points in clusters A, B, C							Centroid Locations		
Iter	0.10	0.15	0.23	0.50	0.60	0.84	0.95	A	B	C
0	-	-	-	-	-	-	-	0.10	0.23	0.84
1: A	1.000	0.716	0.000	0.218	0.140	0.000	0.016	0.134	0.288	0.832
B	0.000	0.280	1.000	0.479	0.255	0.000	0.022			
C	0.000	0.004	0.000	0.302	0.606	1.000	0.961			
2: A	0.965	0.987	0.270	0.193	0.138	0.000	0.020	0.141	0.361	0.843
B	0.033	0.012	0.723	0.575	0.308	0.000	0.030			
C	0.002	0.001	0.007	0.233	0.553	1.000	0.950			
3: A	0.973	0.998	0.675	0.113	0.121	0.000	0.017	0.150	0.493	0.864
B	0.024	0.002	0.311	0.762	0.448	0.000	0.031			
C	0.003	0.000	0.014	0.124	0.431	1.000	0.952			
4: A	0.980	1.000	0.903	0.000	0.046	0.001	0.011	0.157	0.539	0.890
B	0.016	0.000	0.083	0.999	0.820	0.005	0.034			
C	0.004	0.000	0.014	0.000	0.134	0.994	0.955			

If assigned to the highest fuzzy score:

Cluster A: 0.10, 0.15, 0.23

Cluster B: 0.50, 0.60

Cluster C: 0.84, 0.95

The solution is similar to k-means.

14. Consider the following set of one-dimensional data points: {0.1, 0.25, 0.45, 0.55, 0.8, 0.9}. All the points are located in the range between [0,1].

    (a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.4, 1}, respectively, show the cluster assignments and locations of the updated centroids after the first three iterations by filling out the following table.

	Cluster assignment of data points (A, B, or C)						Centroid Locations		
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B	C
0	-	-	-	-	-	-	0.00	0.40	1.00
1									
2									
3									

Calculate the overall sum-of-squared errors of the clustering after the third iteration, where:

$$\text{SSE} = \sum_{x_i \in \text{A}} (x_i - \mu_1)^2 + \sum_{x_i \in \text{B}} (x_i - \mu_2)^2 + \sum_{x_i \in \text{C}} (x_i - \mu_3)^2$$

and μ_js are the centroids of the 3 clusters, A, B, and C.

Answer:

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B	C
0	-	-	-	-	-	-	0.00	0.40	1.00
1	A	B	B	B	C	C	0.10	0.42	0.85
2	A	A	B	B	C	C	0.18	0.5	0.85
3	A	A	B	B	C	C	0.18	0.5	0.85

SSE = 0.0213

(b) For the dataset given in part (a), is it possible to obtain an empty cluster after the first cluster assignment? If possible, what are the locations of the initial centroids to produce the empty cluster? If not, state why.

Answer: Yes. If two of the centroids are randomly initialized to the left of the leftmost points or to the right of the rightmost points. For example, if the initial centroids are (0, 0.05, 0.8), then the first cluster is empty.

(c) Show the clustering results obtained using bisecting kmeans (with k=3). Start by partitioning the data into 2 clusters, with the initial centroids located at 0.10 and 0.90. Then take the cluster with larger SSE and partition it into 2 smaller clusters (you may set the initial centroids to be the two points that are furthest away from each other in the original cluster). Calculate the SSE of the final 3 clusters and compare the results against the k-means solution.

Answer: After the first binary partition:

	Cluster assignment of data points						Centroid Locations	
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B
0	-	-	-	-	-	-	0.10	0.90
1	A	A	A	B	B	B	0.27	0.75
2	A	A	A	B	B	B	0.27	0.75
3	A	A	A	B	B	B	0.27	0.75

The resulting sum-of-squared errors for the clusters are

$$\text{SSE (cluster A)} = 0.0617, \quad \text{SSE (cluster B)} = 0.0650$$

Next, we partition cluster B into 2, denoted as B1 and B2, with initial centroids at 0.55 and 0.90.

	Cluster assignment of data points						Centroid Locations		
Iter	0.10	0.25	0.45	0.55	0.80	0.90	A	B1	B2
0	-	-	-	-	-	-	-	0.55	0.90
1	A	A	A	B1	B2	B2	-	0.55	0.85
2	A	A	A	B1	B2	B2	-	0.55	0.85
3	A	A	A	B1	B2	B2	-	0.55	0.85

The SSE of the bisecting kmeans is 0.0667, which is larger than the SSE for k-means. Therefore, K-means is more effective on this data set. Intuitively, you can see that bisecting K-means always partition the middle cluster into 2 after the first step. So, the middle cluster (which contains the points 0.45 and 0.55) will always be in separate clusters, which is not the optimal solution.

15. K-means is a prototype-based clustering algorithm, in which the choice of the cluster prototype (i.e., representative point) depends on the clustering objective function. For example, consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where each $\mathbf{x}_i$ is a d-dimensional feature vector. The standard k-means algorithm attempts to minimize the following sum-of-square loss function:

$$\min_{\{\boldsymbol{\mu}\}, \mathbf{W}} \sum_{i=1}^{N} \sum_{j=1}^{k} W_{ij} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2, \tag{7.7}$$

where k is the number of clusters, $\boldsymbol{\mu}_j$ is the prototype vector of cluster j, and $\mathbf{W}$ is an $N \times k$ cluster membership matrix ($W_{ij} = 1$ if data point $\mathbf{x}_i$ belongs to cluster j or 0 otherwise). It can be shown that the prototype

vector for the squared loss function is given by the cluster centroid

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} W_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} W_{ik}} \tag{7.8}$$

The denominator term $\sum_{i=1}^{N} W_{ik}$ corresponds to the number of data points assigned to cluster k. Derive the formula for computing the cluster prototype $\boldsymbol{\mu}$ for each scenario described below. Assume W_{ij} is known when computing the prototype. Show your steps clearly.

(a) The spherical k-means algorithm is designed to minimize the following (1 - cosine similarity) loss function:

$$\min_{\{\boldsymbol{\mu}\},\mathbf{W}} \sum_{i=1}^{N}\sum_{j=1}^{k} W_{ij}(1 - \mathbf{x}_i^T\boldsymbol{\mu}_j), \tag{7.9}$$

where each data point $\mathbf{x}_i$ is assumed to have been normalized to unit length, i.e., $\mathbf{x}_i^T\mathbf{x}_i = 1$. The cluster prototypes are also constrained to have unit length, i.e., $\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i = 1$. Hint: solve the constrained optimization problem using the Lagrange multiplier method assuming $\mathbf{W}$ is fixed.

Answer: The Lagrangian formulation for the constrained optimization problem can be written as follows

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{k} W_{ij}(1 - \mathbf{x}_i^T\boldsymbol{\mu}_j) + \sum_{j=1}^{k}\lambda_j(\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j - 1)$$

Taking its partial derivative with respect to $\boldsymbol{\mu}_m$ and setting it to zero yields

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_m} &= -\sum_{i=1}^{N} W_{im}\mathbf{x}_i + 2\lambda_m \boldsymbol{\mu}_m = 0 \\
&\Longrightarrow \quad \boldsymbol{\mu}_m = \frac{\sum_{i=1}^{N} W_{im}\mathbf{x}_i}{2\lambda_m} \\
&\Longrightarrow \quad \boldsymbol{\mu}_m^T \boldsymbol{\mu}_m = 1 = \frac{1}{4\lambda_m^2} \sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j \\
&\Longrightarrow \quad \lambda_m = \frac{1}{2}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j} \\
&\Longrightarrow \quad \boldsymbol{\mu}_m = \frac{\sum_{i=1}^{N} W_{im}\mathbf{x}_i}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W_{im}W_{jm}\mathbf{x}_i^T\mathbf{x}_j}}
\end{aligned}$$

(b) The standard k-means clustering algorithm assumes each point is equally important in determining the clusters. In practice, some data points may be outliers or noise. One way to address this would be to assign a weight to each point, giving lower weights for points that are considered outliers or noise (assume the outliers and noise points were already pre-determined during preprocessing) and higher weights to other legitimate points. Let $0 \leq v_i \leq 1$ be the weight for a data point $\mathbf{x}_i$. The loss function for the "weighted k-means" algorithm is given as follows:

$$\min_{\{\boldsymbol{\mu}\},\mathbf{W}} \sum_{i=1}^{N}\sum_{j=1}^{k} v_i W_{ij} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \tag{7.10}$$

Answer: Taking the partial derivative of the objective function with respect to $\boldsymbol{\mu}_k$ and setting it to zero yields

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} &= \sum_{i=1}^{N} v_i W_{ik}\Big[-2\mathbf{x}_i + 2\boldsymbol{\mu}_k\Big] = 0 \\
&\Longrightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} v_i W_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} v_i W_{ik}}
\end{aligned}$$

which is similar to the original k-means approach except the centroids are computed based on a weighted average of the data points associated with the cluster.

7.2 Cluster Validation

1. Consider the confusion matrices for two clustering solutions as shown below, where the rows correspond to the clusters and the columns correspond to the ground truth classes. Note that solution 2 simply partitions the first cluster of solution 1 into two smaller sub-clusters.

	Solution 1	
	Ground truth class	
	Class 1	Class 2
Cluster 1	40	20
Cluster 2	10	30

	Solution 2	
	Ground truth class	
	Class 1	Class 2
Cluster 1	35	15
Cluster 2	5	5
Cluster 3	10	30

Each entry n_{ij} in the matrix corresponds to the number of data points assigned to cluster i that belong to class j. Furthermore, let $n_{i+} = \sum_j n_{ij}$ (i.e., the sum of all entries in row i) be the number of points in cluster i, $n_{+j} = \sum_i n_{ij}$ (i.e., the sum of all entries in column j) be the number of data points that belong to class j, and $N = \sum_{ij} n_{ij}$ (i.e., the sum of all entries in the table) be the total number of data points. In this exercise, you will compare the performance of the two clustering solutions using the following measures:

- Entropy, $e = \sum_i \frac{n_{i+}}{N} e_i$, where $e_i = -\sum_j \frac{n_{ij}}{n_{i+}} \log \frac{n_{ij}}{n_{i+}}$ is the entropy of cluster i
- Purity, $p = \sum_i \frac{n_{i+}}{N} p_i$, where $p_i = \max_j \frac{n_{ij}}{n_{i+}}$ is the purity of cluster i.
- Normalized mutual information

$$NMI = \frac{2 \sum_{i,j} \frac{n_{ij}}{N} \log \frac{n_{ij} N}{n_{i+} n_{+j}}}{H_1 + H_2},$$

where $H_1 = -\sum_i \frac{n_{i+}}{N} \log \frac{n_{i+}}{N}$, and $H_2 = -\sum_j \frac{n_{+j}}{N} \log \frac{n_{+j}}{N}$.

Answer the following questions:

(a) Compute the values of entropy, purity, and NMI when the clusters are pure (i.e., contains only data points from one class). Assume number of clusters is the same as number of classes (i.e., $k = 2$).

Answer:

$$Entropy = 0$$

$$Purity = 1$$

$$NMI = 1$$

(b) Compute the entropy for both solutions. Which solution is better?

Answer:

$$e_{solution1} = 0.8755$$

$$e_{solution2} = 0.8652$$

Hence, solution 2 is better.

(c) Compute the purity for both solutions. Which solution is better?

Answer:

$$p_{solution1} = 0.7$$

$$p_{solution2} = 0.7$$

Both solutions are equivalent good.

(d) Compute the NMI for both solutions. Which solution is better?

Answer:

$$NMI_{solution1} = 0.1263$$

$$NMI_{solution2} = 0.1142$$

Solution 1 is better

(e) Based on your answers above, state which supervised measure do you think is better and why?

Answer:

Solution 2 has an extra cluster but the clusters are not much better than those in solution 1. This is because entropy tends to be biased towards larger number of clusters. Thus, NMI is the better measure as it is not biased by the number of clusters.

2. Consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ that contains N points, where each data point $\mathbf{x}_i$ is a d-dimensional vector of continuous-valued attributes. Let $\boldsymbol{\mu}_0$ denote the global centroid of the data, i.e.,

$$\boldsymbol{\mu}_0 = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}$$

The total sum-of-squared error assuming there is only 1 cluster is given by

$$\text{TSS} = \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}_0\|_2^2 = \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu}_0)^T (\mathbf{x}_i - \boldsymbol{\mu}_0),$$

where $\|\mathbf{z}\|_2^2 = \mathbf{z}^T\mathbf{z} = \sum_{j=1}^{d} z_j^2$.

(a) Suppose the data is partitioned into 2 clusters, C_1 and C_2, using k-means. Let $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ be the centroids of the two clusters found. Furthermore, let n_1 and n_2 be the number of points in clusters 1 and 2, respectively, where $n_1 + n_2 = N$. The relationship between $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$, and $\boldsymbol{\mu}_2$ is given by

$$\boldsymbol{\mu}_0 = w_1\boldsymbol{\mu}_1 + w_2\boldsymbol{\mu}_2$$

Derive the mathematical expressions for w_1 and w_2 in terms of n_1, n_2, and N.

Answer:

$$\begin{aligned}
\boldsymbol{\mu}_0 &= \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N} \\
&= \frac{\sum_{x_i \in C_1} \mathbf{x}_i + \sum_{x_i \in C_2} \mathbf{x}_i}{N} \\
&= \frac{n_1\boldsymbol{\mu}_1 + n_2\boldsymbol{\mu}_2}{N} \quad \text{since } \boldsymbol{\mu}_j = \frac{\sum_{x_i \in C_j} \mathbf{x}_i}{n_j} \\
&= \frac{n_1}{N}\boldsymbol{\mu}_1 + \frac{n_2}{N}\boldsymbol{\mu}_2
\end{aligned}$$

(b) The sum-of-squared errors for k-means algorithm with 2 clusters, C_1 and C_2, is given by

$$\text{SSE} = \sum_{\mathbf{x}_i \in C_1} \|\mathbf{x}_i - \boldsymbol{\mu}_1\|_2^2 + \sum_{\mathbf{x}_i \in C_2} \|\mathbf{x}_i - \boldsymbol{\mu}_2\|_2^2$$

Show that TSS $\geq$ SSE, i.e., sum-of-squared errors for 2 clusters is always smaller than or equal to the sum-of-squared errors with only 1 cluster.

Answer:

$$\begin{aligned}
\text{SSE} &= \sum_{\mathbf{x}_i \in C_1} \|\mathbf{x}_i - \boldsymbol{\mu}_1\|_2^2 + \sum_{\mathbf{x}_i \in C_2} \|\mathbf{x}_i - \boldsymbol{\mu}_2\|_2^2 \\
&= \sum_{\mathbf{x}_i \in C_1} \|\mathbf{x}_i - \boldsymbol{\mu}_0 + \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2^2 + \sum_{\mathbf{x}_i \in C_2} \|\mathbf{x}_i - \boldsymbol{\mu}_0 + \boldsymbol{\mu}_0 - \boldsymbol{\mu}_2\|_2^2 \\
&= \sum_{\mathbf{x}_i \in C_1} \Big(\|\mathbf{x}_i - \boldsymbol{\mu}_0\|_2^2 + \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2^2 + 2(\mathbf{x}_i - \boldsymbol{\mu}_0)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \Big) \\
&+ \sum_{\mathbf{x}_i \in C_2} \Big(\|\mathbf{x}_i - \boldsymbol{\mu}_0\|_2^2 + \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2\|_2^2 + 2(\mathbf{x}_i - \boldsymbol{\mu}_0)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2) \Big) \\
&= \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}_0\|_2^2 + n_1 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2^2 + n_2 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2\|_2^2 \\
&+ 2 \sum_{\mathbf{x}_i \in C_1} (\mathbf{x}_i - \boldsymbol{\mu}_0)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) + 2 \sum_{\mathbf{x}_i \in C_2} (\mathbf{x}_i - \boldsymbol{\mu}_0)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2)
\end{aligned}$$

Since

$$\sum_{\mathbf{x}_i \in C_j} (\mathbf{x}_i - \boldsymbol{\mu}_0) = n_j(\boldsymbol{\mu}_j - \boldsymbol{\mu}_0) = -n_j(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_j),$$

we have

$$\begin{aligned}
\text{SSE} &= \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}_0\|_2^2 + n_1 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2^2 + n_2 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2\|_2^2 \\
&- 2n_1(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - 2n_2(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2)^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2) \\
&= \text{TSS} - n_1 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2^2 - n_2 \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_2\|_2^2
\end{aligned}$$

Since $\|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_j\|_2^2 \geq 0$, therefore SSE $\leq$ TSS.

3. Consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ that contains N points, where each data point $\mathbf{x}_i$ is a p-dimensional vector of continuous-valued attributes. Suppose the N data points are grouped into two clusters, C_1 and C_2 using k-means clustering. Show that the SSE is non-increasing when the data is split (from 1 cluster containing all N points) into 2 clusters.

Answer: For this proof, we use the following property of a vector: $\|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v} \geq 0$. The SSE for 1 cluster is given by

$$\text{SSE}_1 = \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{c}\|^2 = \sum_{i=1}^{N} \Big[\mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{c}^T\mathbf{x}_i + \mathbf{c}^T\mathbf{c}\Big], \quad (7.11)$$

where

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \quad \text{or} \quad \sum_{i=1}^{N}\mathbf{x}_i = N\mathbf{c}.$$

Thus, Equation (7.11) can be simplified as

$$\text{SSE}_1 = \sum_{i=1}^{N}\mathbf{x}_i^T\mathbf{x}_i - 2N\mathbf{c}^T\mathbf{c} + N\mathbf{c}^T\mathbf{c} = \sum_{i=1}^{N}\mathbf{x}_i^T\mathbf{x}_i - N\mathbf{c}^T\mathbf{c} \quad (7.12)$$

Suppose the data points are split into 2 clusters, $\mathcal{C}_1$ and $\mathcal{C}_2$. The total SSE for the 2 clusters is given by

$$\begin{aligned}
\text{SSE}_2 &= \sum_{\mathbf{x}_i \in \mathcal{C}_1} \|\mathbf{x}_i - \mathbf{c}_1\|^2 + \sum_{\mathbf{x}_i \in \mathcal{C}_2} \|\mathbf{x}_i - \mathbf{c}_2\|^2 \\
&= \sum_{\mathbf{x}_i \in \mathcal{C}_1} \mathbf{x}_i^T\mathbf{x}_i - N_1\mathbf{c}_1^T\mathbf{c}_1 + \sum_{\mathbf{x}_i \in \mathcal{C}_2} \mathbf{x}_i^T\mathbf{x}_i - N_2\mathbf{c}_2^T\mathbf{c}_2 \\
&= \sum_{i=1}^{N}\mathbf{x}_i^T\mathbf{x}_i - N_1\mathbf{c}_1^T\mathbf{c}_1 - N_2\mathbf{c}_2^T\mathbf{c}_2 \qquad (7.13)
\end{aligned}$$

Let

$$\Delta = \text{SSE}_1 - \text{SSE}_2 = N_1\mathbf{c}_1^T\mathbf{c}_1 + N_2\mathbf{c}_2^T\mathbf{c}_2 - N\mathbf{c}^T\mathbf{c}.$$

Next, we can express $\mathbf{c}$ in terms of N_1, N_2, $\mathbf{c}_1$ and $\mathbf{c}_2$. Note that

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i = \frac{1}{N}\Big[\sum_{\mathbf{x}_i \in \mathcal{C}_1}\mathbf{x}_i + \sum_{\mathbf{x}_i \in \mathcal{C}_2}\mathbf{x}_i\Big] = \frac{1}{N}\Big(N_1\mathbf{c}_1 + N_2\mathbf{c}_2\Big)$$

Thus,

$$\begin{aligned}
\Delta &= N_1\mathbf{c}_1^T\mathbf{c}_1 + N_2\mathbf{c}_2^T\mathbf{c}_2 - \frac{N}{N^2}\Big(N_1\mathbf{c}_1 + N_2\mathbf{c}_2\Big)^T\Big(N_1\mathbf{c}_1 + N_2\mathbf{c}_2\Big) \\
&= \frac{1}{N}\Big(NN_1\mathbf{c}_1^T\mathbf{c}_1 + NN_2\mathbf{c}_2^T\mathbf{c}_2\Big) - \frac{1}{N}\Big(N_1^2\mathbf{c}_1^T\mathbf{c}_1 + N_2^2\mathbf{c}_2^T\mathbf{c}_2 + 2N_1N_2\mathbf{c}_1^T\mathbf{c}_2\Big) \\
&= \frac{1}{N}\Big[N_1(N-N_1)\mathbf{c}_1^T\mathbf{c}_1 + N_2(N-N_2)\mathbf{c}_2^T\mathbf{c}_2 - 2N_1N_2\mathbf{c}_1^T\mathbf{c}_2\Big] \\
&= \frac{1}{N}\Big[N_1N_2\mathbf{c}_1^T\mathbf{c}_1 + N_2N_1\mathbf{c}_2^T\mathbf{c}_2 - 2N_1N_2\mathbf{c}_1^T\mathbf{c}_2\Big] \\
&= \frac{N_1N_2}{N}\|\mathbf{c}_1 - \mathbf{c}_2\|^2 \\
&\geq 0
\end{aligned}$$

since $\|\cdot\|^2$ is always non-negative. Thus $\text{SSE}_1 \geq \text{SSE}_2$, which completes the proof.

8

Alternative Clustering

8.1 Type of Clustering

1. State the type of clustering for each method given below. You need to indicate whether it is: (a) partitional (non-hierarchical) or hierarchical, (b) exclusive (disjoint), overlapping, or fuzzy, (c) partial or complete.

   Example: K-means clustering: partitional, exclusive, complete.

   (a) Spectral clustering

   Answer: partitional, exclusive, complete.

   (b) Single link

   Answer: hierarchical, exclusive (for each level of the hierarchy), complete.

   (c) Group average method

   Answer: hierarchical, exclusive (for each level of the hierarchy), complete.

   (d) DBScan.

   Answer: partitional, exclusive, partial.

8.2 HierarchicalClustering

1. Use the distance matrix shown in the table below to perform single and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged and the y-axis show the distance between pairs of clusters being merged at each iteration.

	p1	p2	p3	p4	p5
p1	0	0.3830	0.1474	0.3466	0.1616
p2	0.3830	0	0.4592	0.1252	0.2227
p3	0.1474	0.4592	0	0.3840	0.2643
p4	0.3466	0.1252	0.3840	0	0.2102
p5	0.1616	0.2227	0.2643	0.2102	0

Answer:

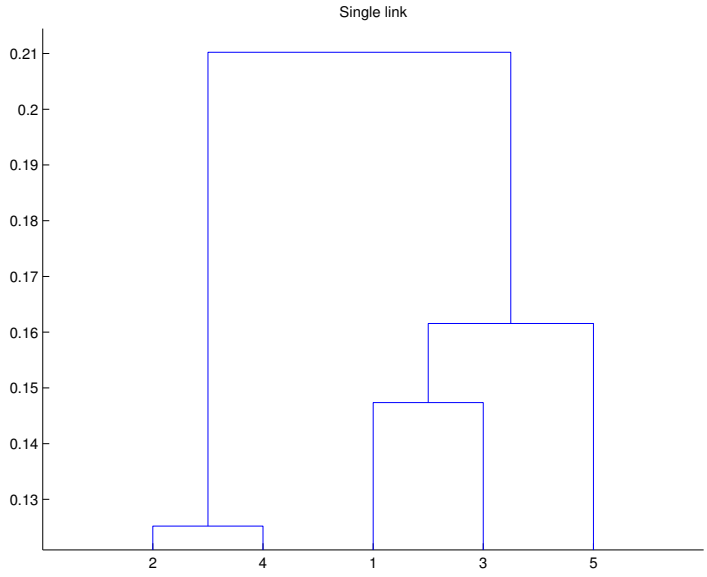


Figure 8.1. Single link.

2. Consider the five objects shown in Figure 8.3.

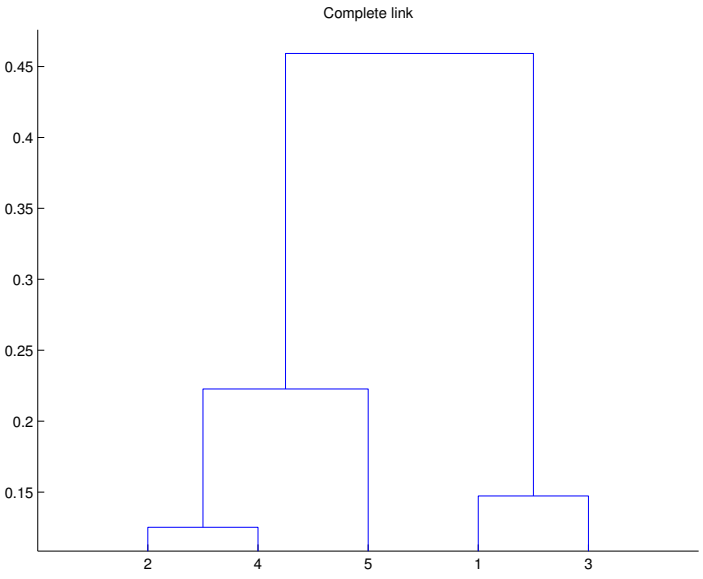


Figure 8.2. Complete link.

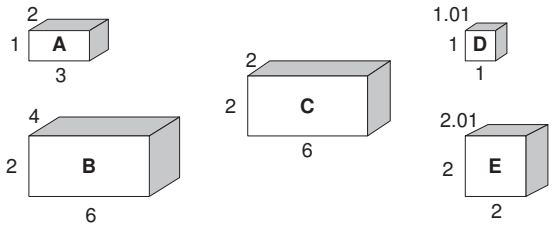


Figure 8.3. Data set for question 2.

(a) Suppose we apply the single link (MIN) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the similarity measure is Euclidean distance.

Answer: Euclidean distance matrix

	A	B	C	D	E
A	0	3.74	3.16	2.23	1.41
B	3.74	0	2.00	5.91	4.47
C	3.16	2.00	0	5.19	4.00
D	2.23	5.91	5.19	0	1.73
E	1.41	4.46	4.00	1.73	0

The dendrogram is shown in Figure 8.4.

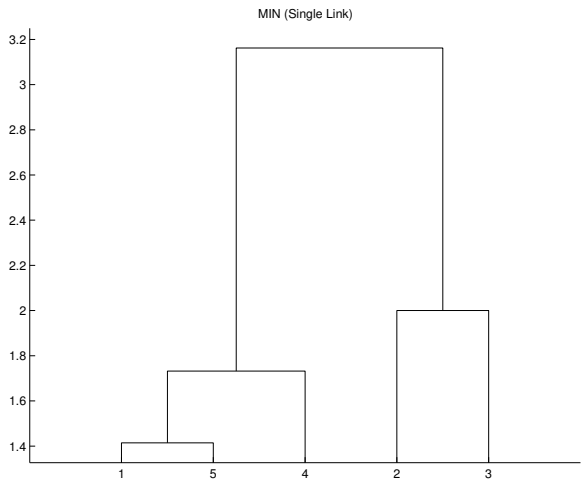


Figure 8.4. Dendrogram using single-link with Euclidean distance.

(b) Repeat the question in part (a) assuming that the similarity measure is correlation.

Answer: Correlation matrix

	A	B	C	D	E
A	1	1	0.87	0	0
B	1	1	0.87	0	0
C	0.87	0.87	1	-0.50	-0.50
D	0	0	-0.57	1	1
E	0	0	-0.57	1	1

The dendrogram is shown in Figure 8.5.

(c) Suppose we apply the complete link (MAX) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the similarity measure is Euclidean distance.

Answer:

The dendrogram is shown in Figure 8.6.

(d) Repeat the question in part (c) assuming that the similarity measure is correlation.

Answer:

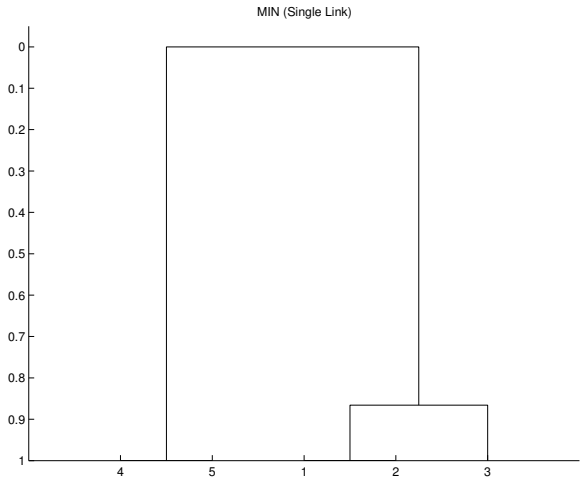


Figure 8.5. Dendrogram using single-link with correlation.

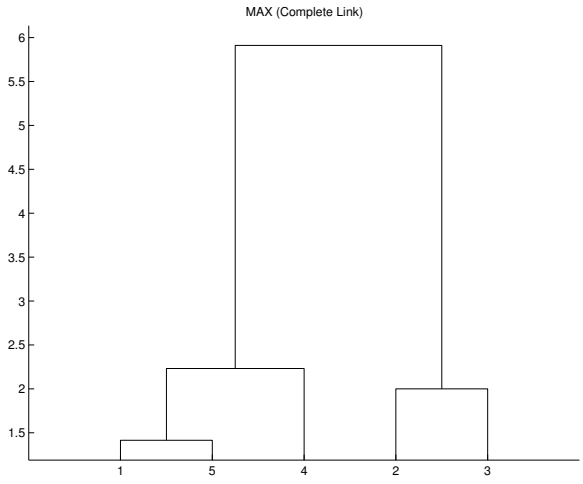


Figure 8.6. Dendrogram using complete-link with Euclidean distance.

The dendrogram is shown in Figure 8.7.

3. Consider the following two-dimensional data points:

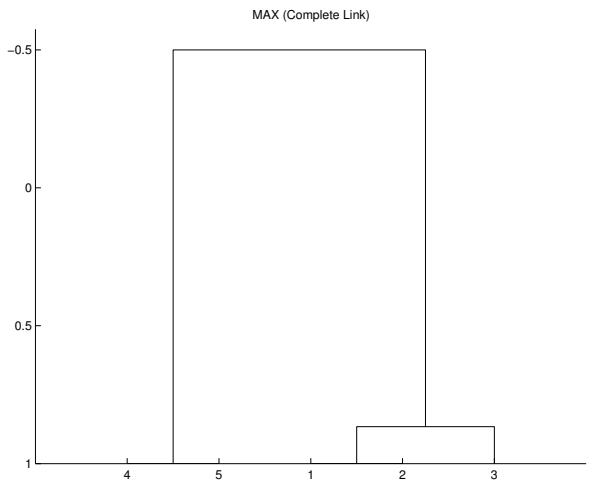


Figure 8.7. Dendrogram using complete-link with correlation.

Data point	x_1	x_2
1	0.1	0.2
2	0.2	0.1
3	0.4	0.8
4	0.5	1.0
5	0.7	0.35

The Euclidean distance between the data points is summarized below:

$$\begin{pmatrix} 0 & 0.1414 & 0.6708 & 0.8944 & 0.6185 \\ 0.1414 & 0 & 0.7280 & 0.9487 & 0.5590 \\ 0.6708 & 0.7280 & 0 & 0.2236 & 0.5408 \\ 0.8944 & 0.9487 & 0.2236 & 0 & 0.6801 \\ 0.6185 & 0.5590 & 0.5408 & 0.6801 & 0 \end{pmatrix}$$

(a) Apply the single link (MIN) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the distance measure is Euclidean. Make sure you label the y-axis of the dendrogram carefully.

Answer: See Figure 8.8.

(b) Suppose we apply the complete link (MAX) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the

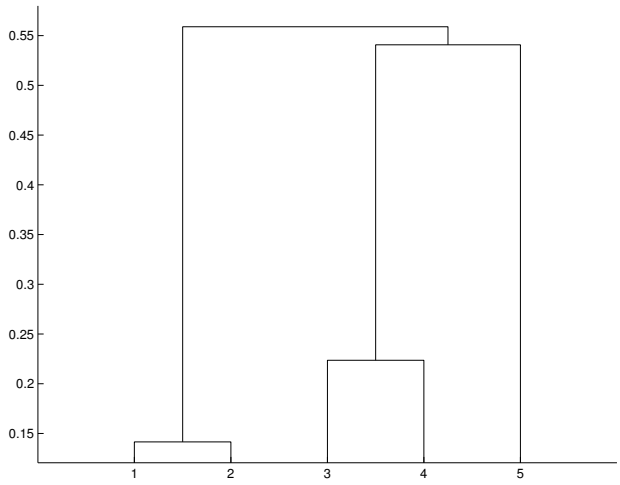


Figure 8.8. Dendrogram for single link (MIN).

distance measure is Euclidean. Make sure you label the y-axis of the dendrogram carefully.

Answer: See Figure 8.9.

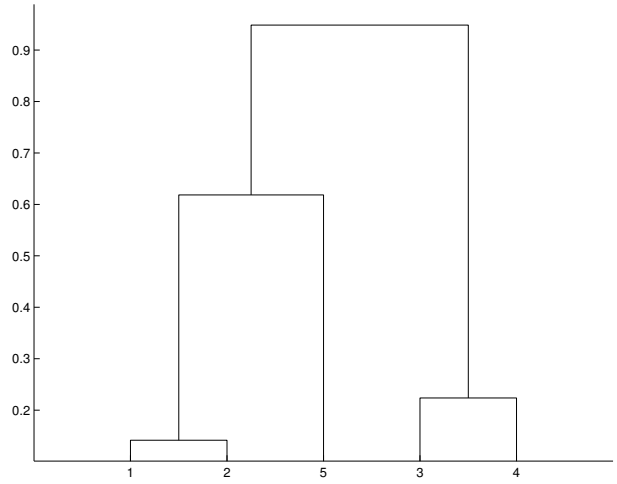


Figure 8.9. Dendrogram for complete link (MAX).

(c) Apply k-means on the data set with $k = 2$. Compare the clustering result against MIN and MAX (with $k = 2$). Which methods produce similar clustering results?

Answer: K-means will put data points 1, 2, and 5 in one cluster, and 3 and 4 in another cluster. The results for MAX and k-means are more similar.

(d) Repeat part (c) using 1 - cosine as distance measure (this approach is also known as spherical k-means). Compare the results against k-means with Euclidean distance, MIN, and MAX algorithms. Explain why the results for spherical k-means are different.

Answer: K-means will put data points 1, 3, and 4 in one cluster, and 2 and 5 in another cluster. These results are different than k-means with Euclidean distance, MIN, and MAX algorithms because the it considers the similarity of angles between the data points instead of their Euclidean distance.

(e) Which method (k-means or spherical k-means) do you think is more appropriate for clustering document data? Why?

Answer: Spherical k-means is more appropriate because Euclidean distance (used in regular k-means) is more sensitive to the document length and inappropriate for asymmetric binary data.

4. Use the distance matrix shown in the table below to perform single and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged and the y-axis show the distance between pairs of clusters being merged at each iteration.

	p1	p2	p3	p4	p5
p1	0	0.5840	0.1955	0.3815	0.1127
p2	0.5840	0	0.6132	0.4956	0.5733
p3	0.1955	0.6132	0	0.2390	0.3067
p4	0.3815	0.4956	0.2390	0	0.4694
p5	0.1127	0.5733	0.3067	0.4694	0

Answer:

Dendrograms are shown in Figure 8.10 and 8.11.

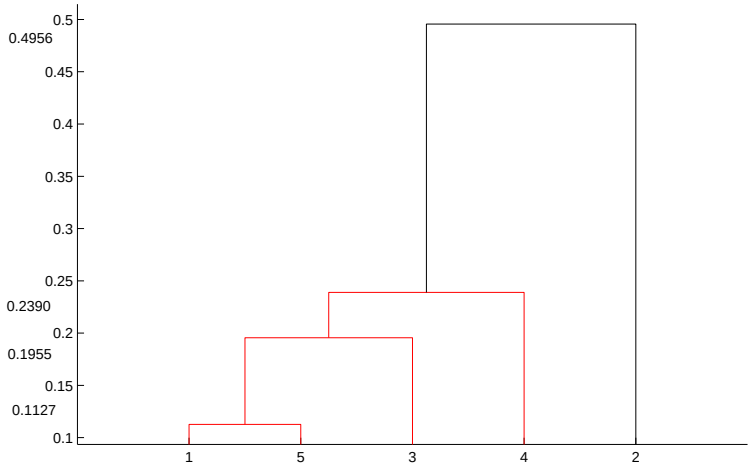


Figure 8.10. Dendrogram for Single Link

5. Consider the following 2-dimensional data set:

Substance	Mass	Volume
A	1	1
B	10	10
C	2	1
D	8	4

(a) Identify the two clusters obtained by using regular k-means algorithm.

Answer: {A,C} and {B,D}

(b) Identify the two clusters obtained by using single-link (MIN) hierarchical clustering.

Answer: {A,C} and {B,D}

(c) Identify the two clusters obtained by using complete-link (MAX) hierarchical clustering.

Answer: {A,C} and {B,D}

(d) How would you modify the k-means algorithm (without modifying the data) so that we obtain two clusters based on the substance density, i.e., we want A and B in one cluster and C and D in another.

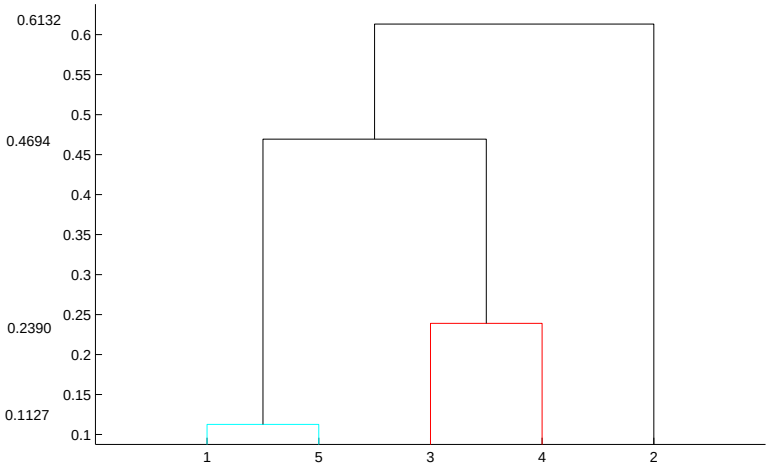
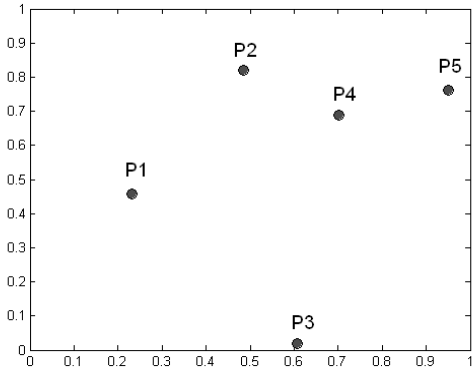


Figure 8.11. Dendrogram for Complete Link

Answer: Use 1 - cosine similarity as distance measure for k-means.

6. Consider the data set shown in Figure 8.12 along with its corresponding distance matrix.



Distance Matrix

	P1	P2	P3	P4	P5
P1	0	0.445	0.577	0.525	0.781
P2	0.445	0	0.812	0.252	0.468
P3	0.577	0.812	0	0.678	0.819
P4	0.525	0.252	0.678	0	0.259
P5	0.781	0.468	0.819	0.259	0.000

Figure 8.12. Data set for hierarchical clustering.

(a) Which of the dendrograms shown in Figure 8.13 corresponds to the clustering solution for single link (MIN) and which corresponds to complete link (MAX)?

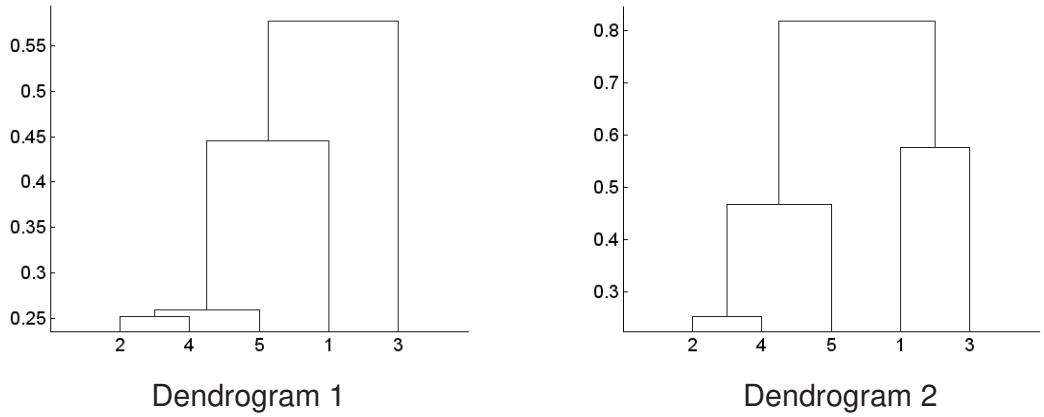


Figure 8.13. Dendrograms.

Answer: Dendrogram 1 is for single link (MIN) and dendrogram 2 is for complete link (MAX).

(b) Show the **cophenetic distance matrix** for single link using the distance matrix given in Figure 8.12.

Answer:

Point	P1	P2	P3	P4	P5
P1	0	0.445	0.577	0.445	0.445
P2	0.445	0	0.577	0.252	0.259
P3	0.577	0.577	0	0.577	0.577
P4	0.445	0.252	0.577	0	0.259
P5	0.445	0.259	0.577	0.259	0

(c) Show the **cophenetic distance matrix** for complete link using the distance matrix given in Figure 8.12.

Answer:

Point	P1	P2	P3	P4	P5
P1	0	0.812	0.577	0.812	0.812
P2	0.812	0	0.812	0.252	0.468
P3	0.577	0.812	0	0.812	0.812
P4	0.812	0.252	0.812	0	0.468
P5	0.812	0.468	0.812	0.468	0

(d) Compute the cophenetic correlation coefficient for the single link and complete link algorithms. Which method is better according to this measure?

Answer: Single link: 0.8191; Complete link: 0.7774

7. Consider the following set of one-dimensional data points:

0.6, 1.2, 1.8, 2.4, 3.0, 4.2, 4.8

(a) Suppose we apply kmeans clustering to obtain two clusters. If the initial centroids are located at 1.8 and 4.5, show the cluster assignments and locations of the centroids after the algorithm converges. Compute the total sum-of-squared errors of the clusters.

Answer: First cluster is 0.6, 1.2, 1.8, 2.4, 3.0.
Error = 3.6
Second cluster is 4.2, 4.8.
Error = 0.18
Total Error = 3.78

(b) Repeat the previous question using 1.5 and 4.0 as the initial centroids. Show the cluster assignment and locations of centroids after the algorithm converges. Compute the total sum-of-squared errors of the clusters.

Answer: First cluster is 0.6, 1.2, 1.8, 2.4 .
Error = 1.8
Second cluster is 3.0, 4.2, 4.8.
Error = 1.68
Total Error = 3.48

(c) What are the two clusters produced by single link?

Answer: The two clusters are {0.6, 1.2, 1.8, 2.4, 3.0} and {4.2, 4.8}.

(d) Which technique, K-means or single link, seems to produce the most natural clustering in this situation? (For K-means, choose the clustering result with the lowest squared error).

Answer: MIN (single link) produces the most natural clustering. Although k-means can produce a similar solution, it was not the clustering with lowest squared error.

8. Consider the following two-dimensional data points:

Data point	x_1	x_2
1	0.1	0.2
2	0.2	0.1
3	0.4	0.8
4	0.5	1.0
5	0.7	0.35

(a) Compute the Euclidean distance between every pair of points. Show your results in a 5×5 distance matrix.

Answer:

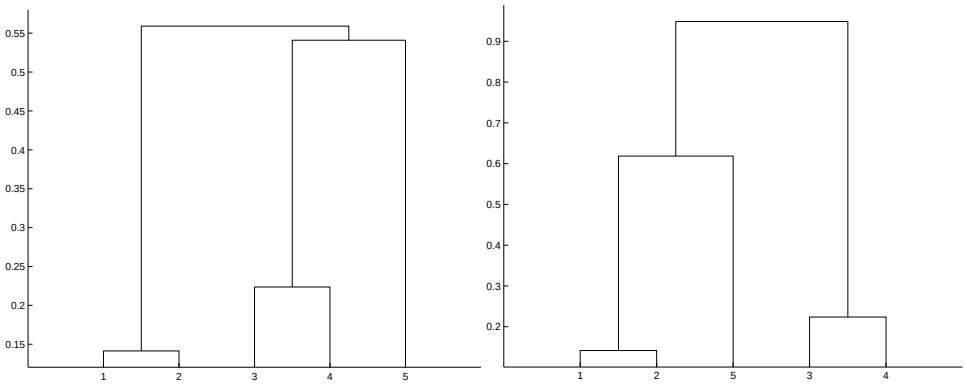
0	0.1414	0.6708	0.8944	0.6185
0.1414	0	0.7280	0.9487	0.5590
0.6708	0.7280	0	0.2236	0.5408
0.8944	0.9487	0.2236	0	0.6801
0.6185	0.5590	0.5408	0.6801	0

(b) Apply the single link (MIN) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the distance measure is Euclidean. Make sure you label the y-axis of the dendrogram carefully.

Answer: See Figure 8.14(a).

(c) Suppose we apply the complete link (MAX) algorithm to cluster the objects. Draw the dendrogram for the clusters assuming the distance measure is Euclidean. Make sure you label the y-axis of the dendrogram carefully.

Answer: See Figure **??**(b).



(a) Single-link clustering (b) Complete-link clustering

Figure 8.14. Hierarchical clustering results.

(d) Apply k-means on the data set with $k = 2$. Compare the clustering result against MIN and MAX (with $k = 2$). Which methods produce similar clustering results?

Answer: The k-means clustering result is shown in Figure 8.15. The two clusters found are {1,2,5} and {3,4}. This is similar to the results of MAX (complete clustering) with $k = 2$.

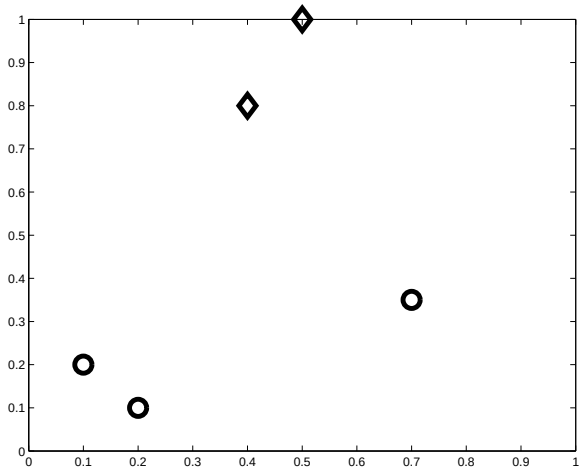


Figure 8.15. K-means clustering.

9. Use the distance matrix shown in the table below to perform single and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged and the y-axis show the distance between pairs of clusters being merged at each iteration.

	p1	p2	p3	p4	p5
p1	0	0.5840	0.1955	0.3815	0.1127
p2	0.5840	0	0.6132	0.4956	0.5733
p3	0.1955	0.6132	0	0.2390	0.3067
p4	0.3815	0.4956	0.2390	0	0.4694
p5	0.1127	0.5733	0.3067	0.4694	0

Answer: The dendrograms for single link and complete link are shown in Figures **??** and **??**, respectively.

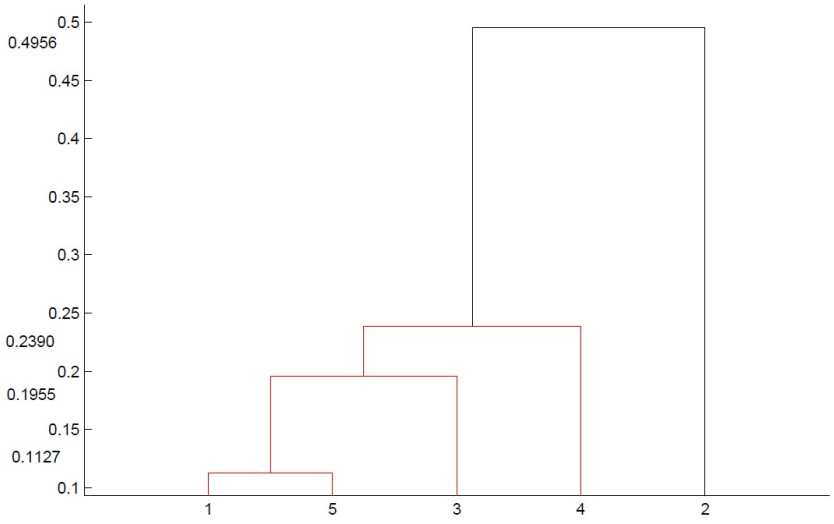


Figure 8.16. Dendrogram for Single Link

10. [4 points] Consider the following Euclidean distance matrix for 5 data points:

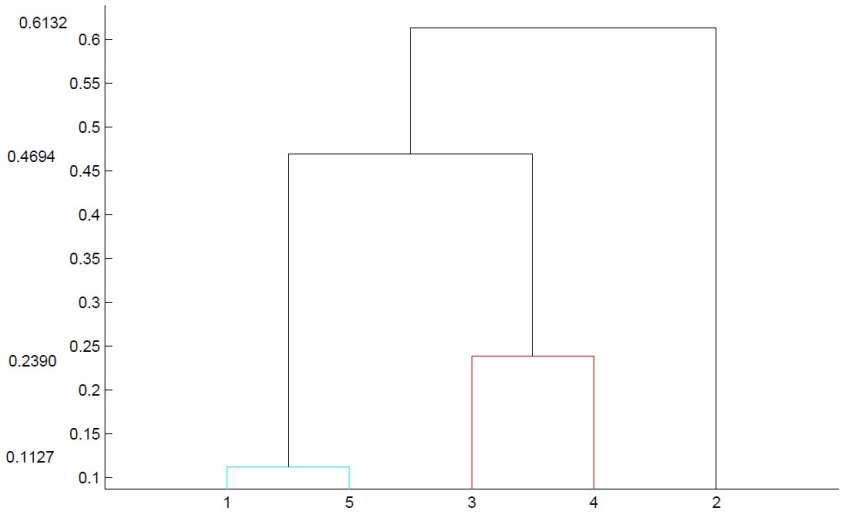


Figure 8.17. Dendrogram for Complete Link

	p1	p2	p3	p4	p5
p1	0	0.4167	0.4974	0.4879	0.3847
p2	0.4167	0	0.7396	0.3413	0.4261
p3	0.4974	0.7396	0	0.5241	0.3439
p4	0.4879	0.3413	0.5241	0	0.1865
p5	0.3847	0.4261	0.3439	0.1865	0

(a) Apply the single link (MIN) algorithm to cluster the objects. Draw the dendrogram for the clusters. Make sure you label the y-axis of the dendrogram carefully.

Answer:

(b) Apply the complete link (MAX) algorithm to cluster the objects. Draw the dendrogram for the clusters. Make sure you label the y-axis of the dendrogram carefully.

Answer:

11. Consider a dataset that has 5 data points, {p1, p2, p3, p4, p5}. Suppose we are interested in applying agglomerative hierarchical clustering to the dataset. The table below shows the distance between every pair of points:

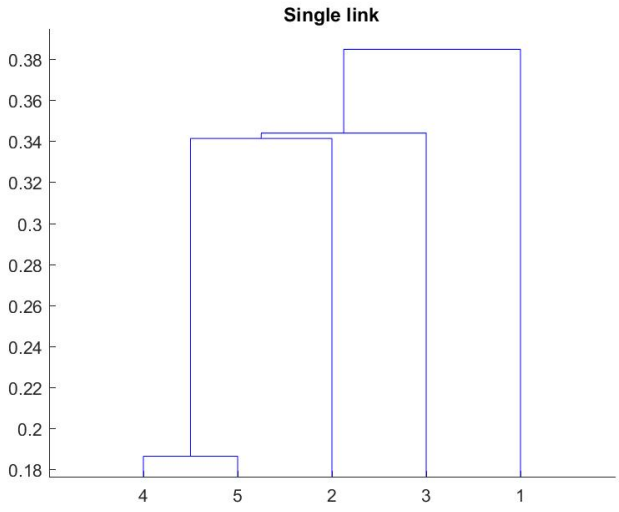


Figure 8.18. Single link results.

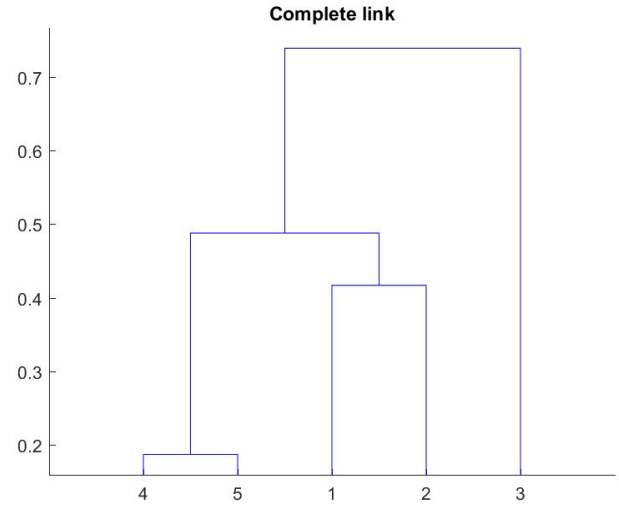


Figure 8.19. Complete link results.

	p1	p2	p3	p4	p5
p1	0	0.8147	0.9058	0.1270	0.9134
p2	0.8147	0	0.6324	0.0975	0.2785
p3	0.9058	0.6324	0	0.5469	0.9575
p4	0.1270	0.0975	0.5469	0	0.9649
p5	0.9134	0.2785	0.9575	0.9649	0

(a) Draw the dendrogram obtained when applying the single link (MIN) clustering method to the dataset. The dendrogram must clearly show the order in which the points are merged. The y-axis of the dendrogram must indicate the distance at which a pair of clusters were merged at each iteration.

Answer: See Figure 8.20.

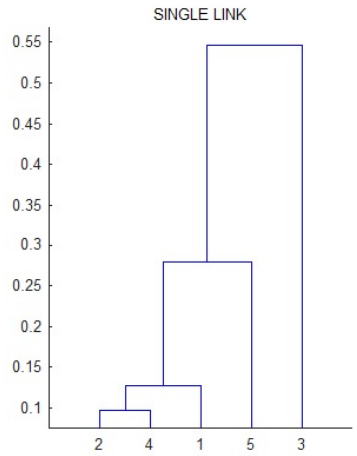


Figure 8.20. Dendrogram for single link

(b) Repeat part (a) by drawing the dendrogram obtained when applying the complete link (MAX) method.

Answer: See Figure 8.21.

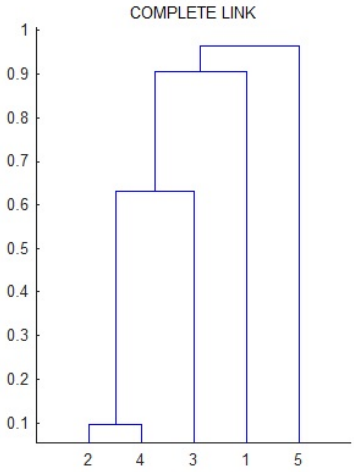


Figure 8.21. Dendrogram for complete link

8.3 Density-based Clustering

1. Consider the data set shown in Figure 8.22. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

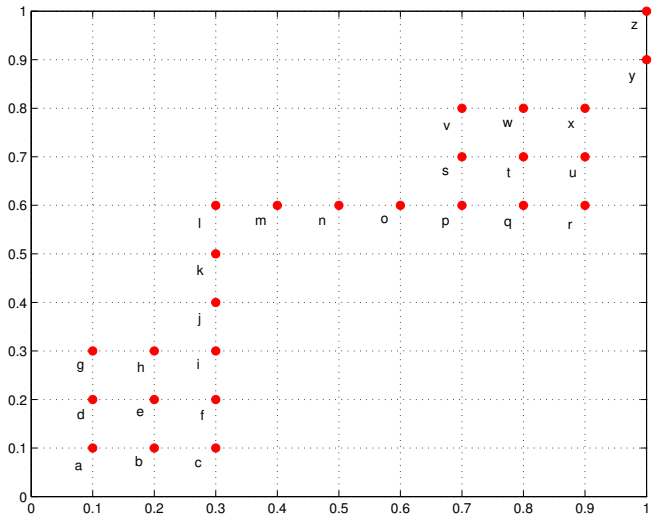


Figure 8.22. DBScan clustering.

(a) List all the core points in the diagram (you can use the labels of the data points in the diagram).

Answer:

MinPts > 3 (including central point): a-p, s, u.

(b) List all the border points in the diagram.

Answer:

MinPts > 3 (including central point): q, r, t, v, x

(c) List all the noise points in the diagram.

Answer:

MinPts > 3 (including central point): w, y, z

(d) Based on the DBScan algorithm described in the book, how many clusters are obtained from the data set?

Answer: 2

2. Consider the data set shown in Figure 8.23. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

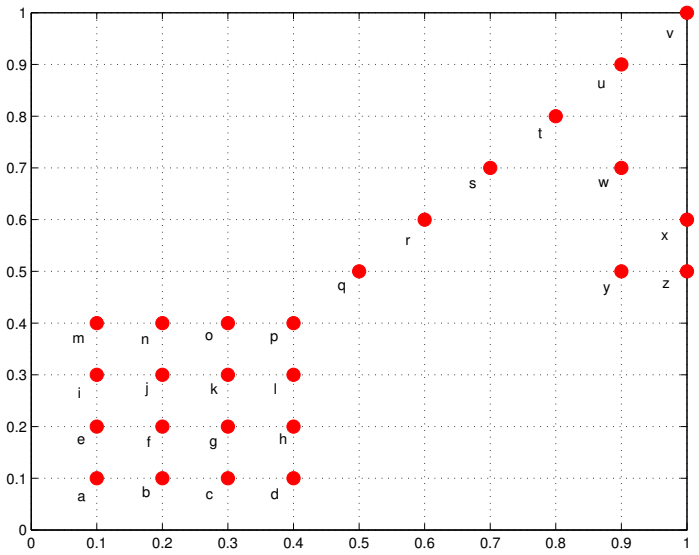


Figure 8.23. DBScan clustering.

(a) List all the core points in the diagram (you can use the labels of the data points in the diagram).

Answer: A point is a core point if the number of points in the Eps neighborhood is more than MinPts (3). So the list of core points are A - P, T, X

(b) List all the border points in the diagram.

Answer: The list of border points are Q, S, U, W, Y, Z

(c) List all the noise points in the diagram.

Answer: The list of noise points are R and V.

(d) Based on the DBScan algorithm, how many clusters are obtained from the data set?

Answer: 3 clusters

3. Consider the data set shown in Figure 8.24. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

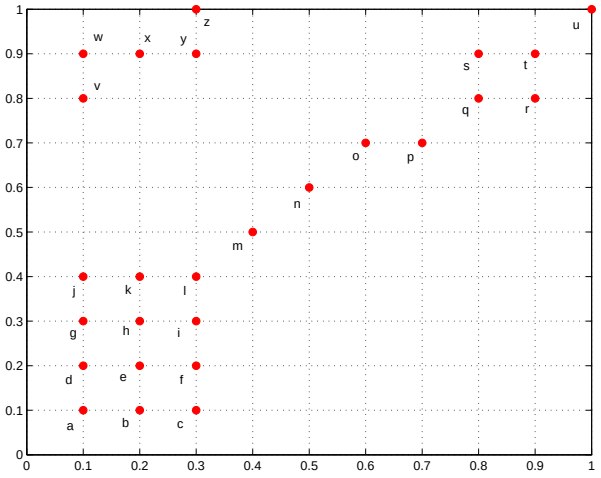


Figure 8.24. DBScan clustering.

(a) List all the core points in the diagram (you can use the labels of the data points in the diagram). Note: a point is considered a core point if there are **more than MinPts** number of points (including the point itself) within a neighborhood of radius Eps.

Answer:

a, b, c, d, e, f, g, h, i, j, k, l, q, r, s, t, x.

(b) List all the border points in the diagram.

Answer:

m, p, u, v, v, w, y, z.

(c) List all the noise points in the diagram.

Answer:

n, o.

(d) Using the DBScan algorithm, how many clusters will be obtained from the data set?

Answer:

3 clusters.

4. Consider the data set shown in Figure **??**. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

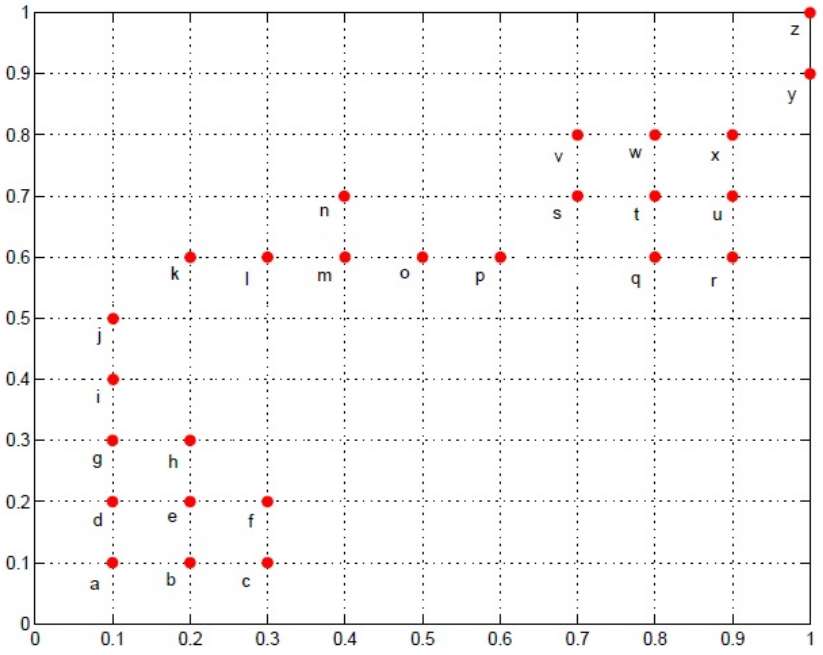


Figure 8.25. DBScan clustering.

(a) List all the core points in the diagram (use the labels a-y from the diagram to indicate which data points are core points). Note: a

point is considered a core point if there are **more than MinPts** number of points (including the point itself) within a neighborhood of radius Eps. Thus, if a data point is in a neighborhood that contains 4 or more points (including itself), it is classified as a core point.

Answer: a-i, l-o, q-x

(b) List all the border points in the diagram.

Answer: j, k, p, y

(c) List all the noise points in the diagram.

Answer: z

(d) Using the DBScan algorithm, what are the clusters obtained from the data set?

Answer: {a - j}, {k - p}, {q - y}.

It is also possible for node p to be assigned to the cluster {q- y}.

5. Consider the data set shown in Figure 8.26. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

(a) List all the core points in the diagram (you can use the labels of the data points in the diagram). Note: a point is considered a core point if there are **more than MinPts** number of points (including the point itself) within a neighborhood of radius Eps.

Answer: a-k, m, o-x

(b) List all the border points in the diagram.

Answer: l, n, y

(c) List all the noise points in the diagram.

Answer: z

(d) Using the DBScan algorithm, how many clusters will be obtained from the data set?

Answer: 2

6. Consider the data set shown in Figure 8.27. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 4.

(a) List all the core points in the diagram (use the labels a-y from the diagram to indicate which data points are core points). Note: a point is considered a core point if there are **more than MinPts**

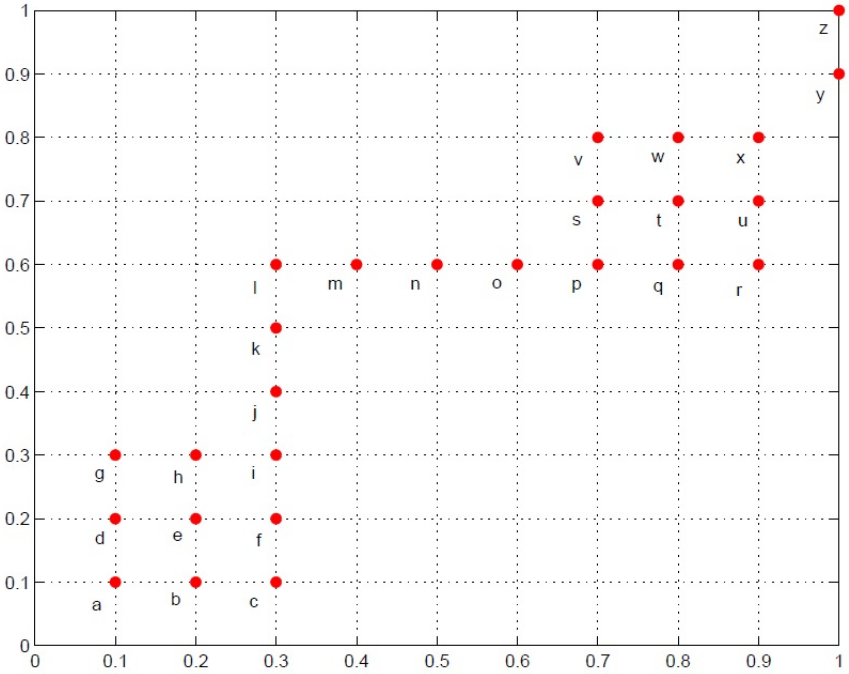


Figure 8.26. Data set for DBScan clustering.

number of points (including the point itself) within a neighborhood of radius Eps. Thus, if a data point is in a neighborhood that contains 5 or more points (including itself), it is classified as a core point.

Answer: b, d, e, f, g, h, i, k, l, s, v.

(b) List all the border points in the diagram.

Answer: a, c, j, m, r, u, t, w.

(c) List all the noise points in the diagram.

Answer: n, o, p, q, x, y, z.

(d) Using the DBScan algorithm, what are the clusters obtained from the data set?

There are 2 clusters found: a - m and r - w.

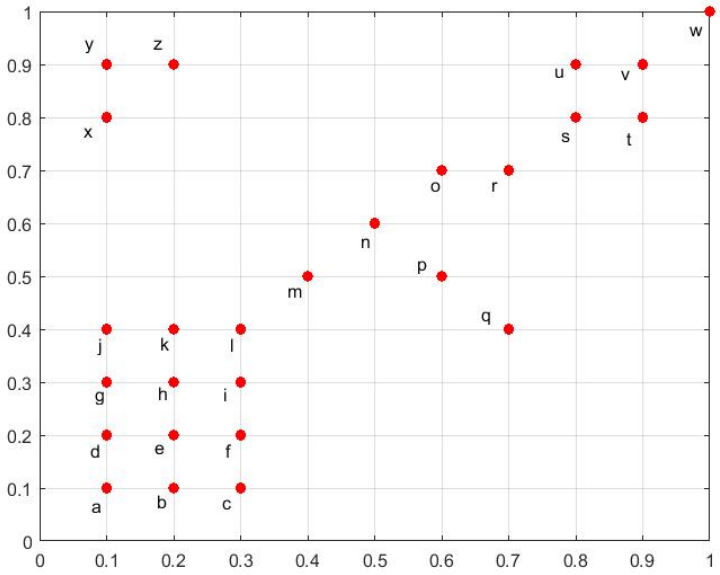


Figure 8.27. DBScan clustering.

8.4 Spectral Clustering

1. Consider the graph data shown in Figure 8.28. Assume the weights for all the links are equal to 1.

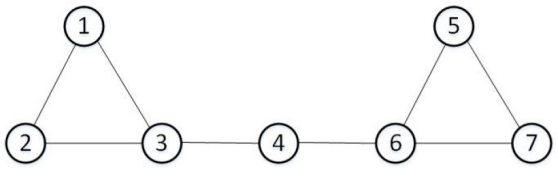


Figure 8.28. Graph data

(a) Compute the Laplacian matrix for the graph. Use the node indices shown in Figure 8.28 to order the rows and columns of the matrix.

Answer:

$$\begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

(b) Compute the first three smallest eigenvalues of the graph Laplacian matrix.

Answer: The three smallest eigenvalues are 0, 0.2679 and 1.5858.

(c) Compute the eigenvectors that correspond to the three smallest eigenvalues given in part (b).

Answer: The eigenvectors are as follows:

$$\begin{pmatrix} 0.3780 & -0.4440 & -0.2808 \\ 0.3780 & -0.4440 & -0.2808 \\ 0.3780 & -0.3251 & 0.1645 \\ 0.3780 & -0.0000 & 0.7941 \\ 0.3780 & 0.4440 & -0.2808 \\ 0.3780 & 0.3251 & 0.1645 \\ 0.3780 & 0.4440 & -0.2808 \end{pmatrix}$$

(d) Apply k-means on the eigenvector matrix to generate 3 clusters. List the three clusters found.

Answer: $V_1 = \{1, 2, 3\}$, $V_2 = \{4\}$, $V_3 = \{5, 6, 7\}$.

(e) Calculate the normalized cut obtained for the 3 clusters found. Let V denote the set of all the nodes in a graph and $W = [w_{ij}]$ denote its adjacency matrix. Suppose V is partitioned into 3 disjoint subsets, V_1, V_2, and V_3, where $V_1 \cup V_2 \cup V_3 = V$. The normalized cut for the partitions can be computed as follows:

$$\text{Ncut}(V_1, V_2, V_3) = \sum_{i=1}^{3} \frac{\text{Cut}(V_i, V - V_i)}{d(V_i)} \tag{8.1}$$

where

$$d(V_i) = \sum_{k \in V_i, j \in V} w_{ij},$$
$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{8.2}$$

Answer: For the solution given in the previous question, $\text{Cut}(V_1, V - V_1) = \text{Cut}(V_3, V - V_3) = 1$, and $\text{Cut}(V_2, V - V_2) = 2$. Furthermore, the degree of the nodes can be found from the diagonal elements of the Laplacian matrix. All the nodes have degree 2 except for data points 3 and 6. Thus, $d(V_1) = 2 + 2 + 3 = 7$, $d(V_2) = 2$, and $d(V_3) = 3 + 2 + 2 = 7$. Hence, the normalized cut for the clusters is

$$\text{Ncut}(V_1, V_2, V_3) = \frac{1}{7} + \frac{2}{2} + \frac{1}{7} = \frac{9}{7}.$$

(f) Suppose the 3 clusters found are as follows:

$$(1, 2), \quad (3, 4, 6), \quad (5, 7)$$

Compute the normalized cut of the clusters. Is the normalized cut smaller, larger, or equal to the solution found in part (d)?

Answer: For this solution, $\text{Cut}(V_1, V - V_1) = \text{Cut}(V_3, V - V_3) = 2$, and $\text{Cut}(V_2, V - V_2) = 4$. Furthermore, $d(V_1) = 2 + 2 = 4$, $d(V_2) = 3 + 2 + 3 = 8$, and $d(V_3) = 2 + 2 = 4$. Hence, the normalized cut for the clusters is

$$\text{Ncut}(V_1, V_2, V_3) = \frac{2}{4} + \frac{4}{8} + \frac{2}{4} = \frac{6}{4}.$$

The normalized cut for this solution is larger than the previous solution.

2. Consider the graph data shown in Figure 8.29. Assume the weights for all the links are equal to 1.

(a) Compute the Laplacian matrix for the graph. Use the node indices shown in Figure 8.29 to order the rows and columns of the matrix.

Answer:

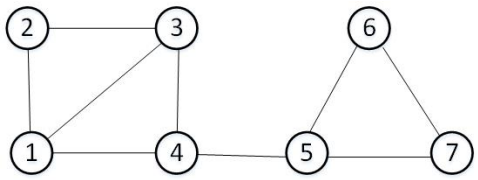


Figure 8.29. Graph data

$$\mathbf{L} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \tag{8.3}$$

(b) Compute the first three smallest eigenvalues of the graph Laplacian matrix.

Answer: 0, 0.3588, 2.2763

(c) Compute the eigenvectors that correspond to the three smallest eigenvalues given in part (b).

Answer:

$$\begin{bmatrix} 0.3780 & -0.3482 & -0.0900 \\ 0.3780 & -0.4244 & 0.6515 \\ 0.3780 & -0.3482 & -0.0900 \\ 0.3780 & -0.1471 & -0.6266 \\ 0.3780 & 0.3078 & -0.2735 \\ 0.3780 & 0.4801 & 0.2143 \\ 0.3780 & 0.4801 & 0.2143 \end{bmatrix}$$

(d) Apply k-means on the eigenvector matrix to generate 3 clusters. List the three clusters found. You may need to run the k-means algorithm multiple times and choose the solution with lowest SSE. For example, you can use Matlab's k-means implementation to do this:

```
matlab> clusters = kmeans( eigv, numClusters, 'Replicates', 200);
```

where eigv corresponds to the matrix of eigenvectors found in part (c), numClusters is the number of desired clusters, and Replicates is the number of times you repeat k-means with different initialization.

Answer: {1,3,4}, {2}, {5,6,7}

(e) Calculate the normalized cut obtained for the 3 clusters found. Let V denote the set of all the nodes in a graph and $W = [w_{ij}]$ denote its adjacency matrix. Suppose V is partitioned into 3 disjoint subsets, V_1, V_2, and V_3, where $V_1 \cup V_2 \cup V_3 = V$. The normalized cut for the partitions can be computed as follows:

$$\text{Ncut}(V_1, V_2, V_3) = \sum_{i=1}^{3} \frac{\text{Cut}(V_i, V - V_i)}{d(V_i)} \tag{8.4}$$

where

$$\begin{aligned} d(V_i) &= \sum_{k \in V_i, j \in V} w_{ij}, \\ \text{Cut}(A, B) &= \sum_{i \in A, j \in B} w_{ij} \end{aligned} \tag{8.5}$$

Answer:

$$\text{Ncut} = \frac{2}{2} + \frac{3}{9} + \frac{1}{7} = 1.4762.$$

3. Consider the graph data shown in Figure 8.30. Assume the weights for all the links are equal to 1.

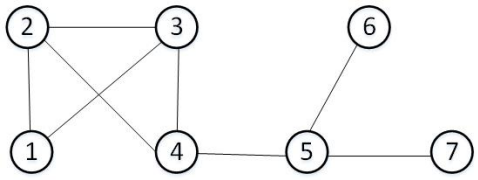


Figure 8.30. Graph data

(a) Compute the Laplacian matrix for the graph. Use the node indices shown in Figure 8.30 to order the rows and columns of the matrix.

Answer:

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

(b) Compute the two smallest eigenvalues of the graph Laplacian matrix.

Answer: 0, 0.3588

(c) Create a 7×2 matrix corresponding to the eigenvectors associated with the two smallest eigenvalues given in part (b).

Answer:

$$\begin{bmatrix} -0.3780 & 0.4244 \\ -0.3780 & 0.3482 \\ -0.3780 & 0.3482 \\ -0.3780 & 0.1471 \\ -0.3780 & -0.3078 \\ -0.3780 & -0.4801 \\ -0.3780 & -0.4801 \end{bmatrix}$$

(d) Apply k-means on the eigenvector matrix to generate 2 clusters. List the two clusters found.

Answer: The two clusters found are {1,2,3,4} and {5,6,7}.

(e) Calculate the normalized cut obtained for the 2 clusters. Let V denote the set of all the nodes in a graph and $\mathbf{W} = [w_{ij}]$ denote its adjacency matrix. Suppose V is partitioned into 2 disjoint subsets, V_1 and V_2, where $V_1 \cup V_2 = V$. The normalized cut for the partitions can be computed as follows:

$$\text{Ncut}(V_1, V_2) = \frac{\text{Cut}(V_1, V_2)}{d(V_1)} + \frac{\text{Cut}(V_1, V_2)}{d(V_2)} \tag{8.6}$$

where

$$\begin{aligned} d(V_i) &= \sum_{k \in V_i, j \in V} w_{kj}, \\ \text{Cut}(V_1, V_2) &= \sum_{i \in V_1, j \in V_2} w_{ij} \end{aligned} \tag{8.7}$$

Answer:

$$\text{Ncut} = \frac{1}{11} + \frac{1}{5} = 0.2909.$$