CSCI-B 565 DATA MINING
Homework 2
Morning Class
Computer Science Core
Fall, Indiana University, Bloomington, IN

Pratish Merchant
pmmercha@iu.edu

December 18, 2015

All the work herein is solely mine

1 k-means Algorithm in Theory

1.1 Solutions

1. No the algorithm doesn't always converge theoretically . It mostly converges on either the global or sometimes the local minima. However , in some cases we do find that it oscillates for some values and is trapped between this values , sometimes slowing down and then again oscillating to another value.However, practical occurrence of the above phenomenon is very rare and we can implement some heuristics which can generally avoid the above condition . However , K-means will not always converge to a global minima , it can get stuck in the local minima.

   To avoid this condition , we may use an extra variable to limit the time spent between values. Lets call this parameter as EscapeLoop, which will be a counter for the amount of time it spends on finding the a particular set of centroids . The value of EscapeLoop will depend on the dataset that is given, the quality of the partitioning required and the time length or the algorithm. It can be in the range of 500 to 1000 in general.

2. The K means algorithm doesn't exactly specify how to initialize the initial variables, which makes it problematic as we don't have a fixed method to initialize the variables. We can discuss a few ways and problems related to them :

   (a) Random Intialization : In this we select centroid with random values . Since we are selecting values at random, we may end up selecting values which are away from all the data and hence , the results will not be optimum and we may end up with empty clusters. This can be overcome by running the algoritm with different random values and selecting the best result.

   (b) Selecting a random point from dataset : This solves the problem having values which are not close to any point. However , we still have the problem of outliers.Using outliers as starting point will make the make the centroid distant from the cluster. It also suffers when centroids randomly selected are very close to each other.

   (c) Select mean points or T distant points from data : We can assign different mean values computed or centroids which are T distant apart from each other, as the initial centroids The major problem with this that the dataset should be ordered and its hard tp decide the value of T for different dataset. Also , all data is not always numbers.

(d) Selecting a farthest point: It is the point which is the farthest away from the first centroid i.e the farthest point from first centroid will become the new centroid and so on . This will have the advantage of spreading the centroids away from each other , but will be susceptible to outliers.

Thus we see that there is no ideal way to initialize variables and each method has its own set of problems. We can combine one or two methods which will reduce th problem , but not completely eliminate it.

3. The runtime of the K means is : $| \triangle | \times | A | \times | EscapeLoop | \times | K |$
where,

$bigtriangleup$ = The dataset A : The number of attributes
EscapeLoop : Parameter to get out of non converging conditions
K : The number of partitions

4. Excluded from homework.

5. (a) For selecting the new centroids , we can use median of a set instead of a direct arithmetic average. We select that points which closely represents the average centroid of the set.

   (b) Select the point whose distance to the previous centroids is the greatest. i.e. the point which is further away from previous selected centroid in that set, would become the next centroid and so on.This will make the centroids cluster in different different directions and well spaced out.

   (c) We can also select the weighted average of the points in the dataset instead of a simple average. This can be done by assigning weights to data inside a cluster. The closer points more weights and farther points less weights. This is nothing bt combination of average and previous method.

   (d) If items in the set are repetitive , we could select the mode of the set i.e. the value which is repeated more than once , if there are any in a set.

6. We can introduce a new variable D_c which should the minimum distance between the centroids selected. The value of D_c would be the three quarter the average distance between the centroids. The average distance can be calculated according to the following function.

Function `checkCenDist`
INPUT (data Δ, distance $d : \Delta^2 \to \mathbb{R}_{\geq 0}$, Set of centroids $\triangle_c$, centoid number k)
OUTPUT (D_c is average value between the centroids)
Assume centroid is structure $c = (v \in DOM(\Delta)$
$c.v$ is the centroid value and $d.B$ is the set of distance between centroids.
$d(c_i, c_j)$ is the Euclidean distance between the two centroids
for $i = 1, k$ **do**
  for $j = 1, k$ **do**
    $d.B = d.B \bigcup d(c_i, c_j)$ ▷ Cal distance between Centroids
  end for
end for
$D_c = \frac{3 * d.B}{4 * k}$
for $i = 1, k$ **do**
  for $j = 1, k$ **do**
    if $(d(c_i, c_j) < D_c)$ centroids_close = true
  end for
end for
Return : *centroids_close*

We can also keep a flag variable centroids_close. This will be true when all the centroids are less than D_c distance value, else it would be false.

The algorithm would be called after line 21 of K means algorithm given in Question 1, which is after the new centroids have been calculated. If the centroids are found too close , centroids_close flag would be true and centroids would be recalculated . Once the flag is false , the loop will break and algorithm would proceed . A snippet of modified code :

$i \leftarrow i + 1$
for $\delta \in \Delta$ **do**
  $c_j^i.B \leftarrow c.B \cup \{\delta\}$, where $\min_{c_j^i}\{d(\delta, c_j^i.v)\}$
  ▷ Associate a data point δ with the nearest centroid $c_j^i.v$
end for
$centoids\_close \leftarrow false$
repeat
  for $j = 1, k$ **do**
    $c_j^i.v \leftarrow ave(c_j^i.B)$ ▷ Update centroid to be *best* representative of nearest data
    $c_j^i.B \leftarrow \emptyset$ ▷ ave is easiest representative
  end for
  checkCenDist(Δ, Set of centroids $\triangle_c$, k)
until $centoids\_close == false$

7. Let $x = \{a, b, c, d\}, y = \{a, b, e\}, z = \{b, f\}, \mathcal{U} = \{a, b, c, d, e, f\}$. Compute the distances using

$$d(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases} \quad (1)$$

The signature of the distance function is: $d : \mathsf{Set}^2 \to \mathbb{R}_{\geq 0}$.

(a) $\neg x = \mathcal{U} - \{a, b, c, d\} = \{e, f\}$

(b) $\neg \mathcal{U} = \emptyset$

(c) $d(x, y) = 1$

(d) $d(x \cap y, \{a, b\}) = 0$

(e) $\mathbf{d}(\mathbf{x}, \mathbf{x} \cup \mathbf{y}) =$

$x \cup y = \{a, b, c, d, e\}$
$d(\{a, b, c, d\}, \{a, b, c, d, e\}) = 1$

(f) $\mathbf{d}(\neg(\mathbf{x} \cap \mathbf{y}), \neg\mathbf{x} \cup \neg\mathbf{y}) =$

$x \cap y = \{a, b\}$
$\neg(x \cap y) = \mathcal{U} - \{a, b\} = \{c, d, e, f\}$

$\neg x = \{e, f\}, \neg y = \{c, d, f\}$
$\neg x \cup \neg y = \{c, d, e, f\}$

$d(\{c, d, e, f\}, \{c, d, e, f\}) = 0$

$$\begin{aligned} J(x, y) &= |x \cap y|/|x \cup y| \\ d(x, y) &= 1 - J(x, y) \end{aligned}$$

The signature of the distance function is: $d : \mathsf{Set}^2 \to \mathbb{R}_{\geq 0}$.

(a) $\mathbf{d(x, y)} =$

$$x \cup y = \{a, b, c, d, e\}, x \cap y = \{a, b\}$$

$$\begin{aligned} J(x, y) &= |x \cap y|/|x \cup y| \\ &= 2/5 \\ d(x, y) &= 1 - J(x, y) \\ &= 1 - 2/5 \\ &= 0.6 \end{aligned}$$

(b) $\mathbf{d(x \cap y, \{a, b\})} =$

$$\begin{gathered} Let\ p = x \cap y = \{a, b\}, q = \{a, b\} \\ p \cap q = \{a, b\}, p \cup q = \{a, b\} \end{gathered}$$

$$\begin{aligned} J(p, q) &= |p \cap q|/|p \cup q| \\ &= 1/1 \\ d(p, q) &= 1 - J(p, q) \\ &= 1 - 1 \\ &= 0 \end{aligned}$$

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

$$\begin{gathered} Let\ p = x = \{a, b, c, d\}, q = x \cup y = \{a, b, c, d, e\} \\ p \cap q = \{a, b, c, d\}, p \cup q = \{a, b, c, d, e\} \end{gathered}$$

$$\begin{aligned} J(p, q) &= |p \cap q|/|p \cup q| \\ &= 4/5 \\ d(p, q) &= 1 - J(p, q) \\ &= 1 - 0.8 \\ &= 0.2 \end{aligned}$$

(d) $\mathbf{d(\neg(x \cap y), \neg x \cup \neg y)} =$

$$\begin{gathered} x \cap y = \{a, b\}\ , \neg x = \{e, f\}, \neg y = \{c, d, f\} \\ \neg(x \cap y) = \mathcal{U} - \{a, b\} = \{c, d, e, f\} \\ \neg x \cup \neg y = \{c, d, e, f\} \\ Let\ p = \neg(x \cap y) = \{c, d, e, f\}, q = \neg x \cup \neg y = \{c, d, e, f\} \\ p \cap q = \{c, d, e, f\}, p \cup q = \{c, d, e, f\} \end{gathered}$$

$$\begin{aligned} J(p, q) &= |p \cap q|/|p \cup q| \\ &= 1/1 \\ d(p, q) &= 1 - J(p, q) \\ &= 1 - 1 \\ &= 0 \end{aligned}$$

8. $\mathbf{x} = 111100$. $\mathbf{y} = 110010, \mathbf{z} = 010001$. The Hamming distance between the vector:

$$c(x,y) = \begin{cases} 0, & x = y \\ 1, & otherwise \end{cases} \text{ for individual characters} \quad (2)$$

$$d(\mathbf{x},\mathbf{y}) = \Sigma_{i=0}^{n} c(\mathbf{x}[i],\mathbf{y}[i]) \quad n = ||\mathbf{x}||, \text{the length of the string.} \quad (3)$$

The signature of the distance function is: $d : \mathsf{String}^2 \to \mathbb{R}_{\geq 0}$. String functions:

$$\textvisiblespace = \text{space} \quad (4)$$

$$concat(\mathtt{b}, \mathtt{at}) = \mathtt{bat} \quad (5)$$

$$concat(\mathtt{bat}, \epsilon) = \mathtt{bat} \quad (6)$$

$$contat(\epsilon, \mathtt{bat}) = \mathtt{bat} \quad (7)$$

$$upper(\mathtt{a}) = \mathtt{A} \quad (8)$$

$$space(\mathtt{b\textvisiblespace a\textvisiblespace t}) = \mathtt{bat} \quad (9)$$

(a) $d(\mathbf{x},\mathbf{y}) = d(111100, 110010) =$

$$\begin{array}{r} 111100 \\ 110010 \\ \hline 001110 \end{array}$$

$d(111100, 110010) = 0+0+1+1+1+0 = 3$

(b) $d(concat(\mathbf{x},\mathbf{x}), concat(\mathbf{x},\mathbf{y})) = d(xx, xy) = d(111100111100, 111100110010)$

$$\begin{array}{r} 111100111100 \\ 111100110010 \\ \hline 000000001110 \end{array}$$

$d(111100111100, 111100110010) = 0+0+0+0+0+0+0+0+1+1+1+0 = 3$

(c) Hamming distance calculates the different in the nature of the strings . It basically is used to count the number of differences in bits (which can be used to calculate error).

If the string supplied is the same , then Hamming distance would be 0.This means that the string sent and the string received is the same and there are no errors in the string . An error is indicated by non zero value.That is why the answer of the first and the second question is identical (3), because we are concatenating the same value x and d(x,x) = 0, which can be looked as the first part is received without errors and there is difference between the other half and hence its different.

In general, performing following operations would result in same hamming distance :
d(concat(xyzyy,x),concat(xyzyy,y))
d(concat(xyz,yyx),concat(xyz,yyy))
d(concat(upper(xyzyy,x)),concat(uper(xyzyy,y)))

(d) $d(\mathtt{N\textvisiblespace orth}, \mathtt{nort\textvisiblespace h}) =$

$$\begin{array}{r} N\textvisiblespace orth \\ nort\textvisiblespace h \\ \hline 111110 \end{array}$$

$d(\mathtt{N\textvisiblespace orth}, \mathtt{nort\textvisiblespace h}) = 1+1+1+1+1+0 = 5$

(e) $d(upper(space(\mathtt{N\textvisiblespace orth})), upper(space(\mathtt{nort\textvisiblespace h}))) =$

$$\begin{array}{ccccc} N & O & R & T & H \\ N & O & R & T & H \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}$$

$d(upper(space(\texttt{N\_orth})), upper(space(\texttt{nort\_h}))) = d(\texttt{NORTH}, \texttt{NORTH}) = 0+0+0+0+0+0 = 0$

(f) $d(\texttt{north}, \texttt{south}) =$

$$\begin{array}{r} north \\ south \\ \hline 10100 \end{array}$$

$d(\texttt{north}, \texttt{south}) = 1+0+1+0+0 = 2$

(g) We can introduce new operations along with the above operations :

$$clean(\texttt{b!a@\~{}\$t.}) = \texttt{bat} \quad (10)$$
$$clip(\texttt{bat}) = \texttt{b} \quad (11)$$
$$clip(\texttt{b}) = \texttt{b} \quad (12)$$

We can write the distance function as : $P(string) = clip(uppper(clean(space(string))))$

$$d(P(string1), P(string2)) = \begin{cases} 0, & x = y \\ 1, & otherwise \end{cases} \quad (13)$$

$$(14)$$

Lets take a few examples :

i. $d(P(N.), P(North))$

$$\begin{aligned} P(N.) &= space(N.) \\ &= clean(N.) \\ &= upper(N) \\ &= clip(N) = N \end{aligned}$$

$$\begin{aligned} P(North) &= space(North) \\ &= clean(North) \\ &= upper(North) \\ &= clip(NORTH) = N \end{aligned}$$

$$d(P(N.), P(North)) = d(N, N) = 1$$

ii. $d(P(\_n), P(nor))$

$$\begin{aligned} P(N.) &= space(\llcorner n) \\ &= clean(n) \\ &= upper(n) \\ &= clip(N) = N \end{aligned}$$

$$\begin{aligned} P(North) &= space(nor) \\ &= clean(nor) \\ &= upper(nor) \\ &= clip(NOR) = N \end{aligned}$$

$$d(P(\llcorner n), P(nor)) = d(N, N) = 1$$

9. Let $\delta = (\mathsf{Set}\ x, \mathsf{String}\ y)$.
where,
$\mathsf{Set}\ x = Set\ of\ 2\ numbers\ [\ a, b\ ]$
$\mathsf{String} = String\ from\ north\ ,\ south\ ,\ east\ ,\ west$

Assumption : The set and string are never null.

$$d((\mathsf{Set}\ x_1, \mathsf{String}\ y_1), (\mathsf{Set}\ x_2, \mathsf{String}\ y_2)) = \begin{cases} 0, & \mathsf{Set}\ x_1 = \mathsf{Set}\ x_2\ ;\ \mathsf{Opposite\ Pair\ Strings} \\ L; & \mathsf{Set}\ x_1 \neq \mathsf{Set}\ x_2\ ;\ \mathsf{Complimentary\ Pair\ Strings} \end{cases}$$

where ,
$L = F(x_2, x_1) + S(y_2, y_1)$
$F(x_2, x_1) = [\ |a_2 - a_1|, |b_2 - b_1|\ ] = a_{diff}\ +\ b_{diff}$
$S(y_2, y_1) =$ sin value of the angle between the directions. For eg :

$$\begin{aligned} sin0 &= sin180\ = 0 \quad \ldots North/South, East/West \\ sin90 &= sin270 = 1 \quad \ldots North/South \leftrightarrow East/West \end{aligned}$$

To illustrate the above we can consider the following example :

$$\begin{aligned} d(([1,2]\ ,\ north), ([3,4]\ ,\ east)) &= L\ =\ d([\ 1,1\ ], sin90)\ =\ 1+1+1=\ 3 \\ d(([23,50]\ ,\ west), ([16,9]\ ,\ east)) &= L\ =\ d([\ 7,41\ ], sin180)\ =\ 7+41+0=\ 48 \\ d(([5,7]\ ,\ west), ([2,8]\ ,\ south)) &= L\ =\ d([\ 3,1\ ], sin90)\ =\ 3+1+1=\ 5 \end{aligned}$$

Proof: $\mathsf{d}((\mathsf{Set}\ x_1, \mathsf{String}\ y_1), (\mathsf{Set}\ x_2, \mathsf{String}\ y_2))$ is a metric

$\forall\, x, d(x, x) = 0.$
Let tuple be $x = \delta$([1,2],north)
$d(([1,2], north), [1,2], north) = 0$ by definition.
Let tuple be $x = \delta$ ([3,4],south)
$d(([3,4], south), [3,4], south) = 0$ by definition.

$\forall\, x, y\ d(x, y) = d(y, x)$
Let $x = \delta([1,2], north)\ ,\ y = \delta([3,4], east)$ be any two unequal values.

According to defination of d , since tuples not equal
$d(x,y) = d(([1,2], north), ([3,4], east)) = L = d([1,1], sin90) = 1+1+1 = 3$
$d(y,x) = d(([3,4], east), ([1,2], north)) = L = d([1,1], sin90) = 1+1+1 = 3$
$\therefore \; d(x,y) = d(y,x)$

$\forall\, x, y, z \; d(x,y) + d(y,z) \geq d(x,z)$
Let $x = \delta([1,2], north)$, $y = \delta([3,4], east)$, $\delta([7,2], south)$ be any three values such that $a \neq b \neq c$.
$\therefore$ According to definition ,
$d(x,y) = d(([1,2], north), ([3,4], east)) = L = ([2,2], sin90) = 2+2+1 = 5$
$d(y,z) = d(([3,4], east), ([7,2], south)) = L = ([4,2], sin90) = 4+2+1 = 7$
$d(x,z) = d(([1,2], north), ([7,2], south)) = L = ([6,0], sin180) = 6+0+0 = 6$
Hence, $d(a,b) + d(b,c) \geq d(a,c)$ . . . (Since $(7+5) \geq 6$)

Let $x = \delta([1,0], north)$, $y = \delta([1,1], south)$, $\delta([0,1], south)$ be any three values such that $x \neq y \neq zc$.
$\therefore$ According to definition ,
$d(x,y) = d(([1,0], north), ([1,1], south)) = L = ([0,1], sin180) = 0+1+0 = 1$
$d(y,z) = d(([1,1], south), ([0,1], south)) = L = ([1,0], sin0) = 1+0+0 = 1$
$d(x,z) = d(([1,0], north), ([0,1], south)) = L = ([1,1], sin180) = 1+1+0 = 2$
Hence, $d(a,b) + d(b,c) \geq d(a,c)$ . . . (Since $(1+1) \geq 2$)

2 Application of k-means to medical data

2.1 Summary and Assumption

- The data provided is of patients who have performed biopsies.
- Since the dataset Δ is having some missing and duplicated values , we clean the data to find reduced no of tuples : Δ_{clean} having 675 tuples (16 missing values and 8 duplicate SCN rows).
- We assume that a person can have only one unique SCN number associated with himself.
- Patient would have different values of all the attributes (thickness , cell size ... etc.) ,if the patient has performed biopsy more than once. Hence, only rows with exact same values of attributes corresponding to a particular SCN would be considered as duplicate.This assumption follows the logic that biopsy is very costly and a person will not have more than few biopsies.
- The missing data value is only for column Bare Nuclei, but the classification of these patients into benign and malignant cancer is known. These values will be considered for evaluating the costs.
- A patient might perform Mastectomy when he is diagnosed with a malignant cancer class(4).
- The attribute to column mapping is as shown below :

Column Name	Attribute
Thickness	(A_1)
Cell Size	(A_2)
Cell Shape	(A_3)
Marginal Adhesion	(A_4)
Single Epithelial CS	(A_5)
Bare Nuclei	(A_6)
Bland Chromatin	(A_7)
Normal Nucleoli	(A_8)
Mitoses	(A_9)
Class	(A_{10})

1. Few considerations for evaluating the data :
   - We dont consider the duplicate rows for evaluating the biopsy cost. The total unique records are 691 , of which 238 are malignant.
   - A patient might perform Mastectomy when he is diagnosed with a malignant cancer class(4).

   (a) The total cost(TC) of biopsies would be

   $$TC = Number\ of\ records\ from\ clean\ data * Cost\ of\ Biopsy$$
   $$TC = 691 * (Min\ : \$1000\ ,\ Avg\ :\ \$3000\ ,\ Max\ :\ \$5000)$$
   $$TC = Min\ : \$691,000\ ,\ Avg\ :\ \$2,073,000\ ,\ Max\ :\ \$3,455,000$$

   (b) The total cost (TC) of mastectomies would be

   $$TC = Number\ of\ malignant\ records\ from\ clean\ data * cost\ of\ Mastectomy$$
   $$TC = 238 * (Min\ : \$15000\ ,\ Avg\ :\ \$35000\ ,\ Max\ :\ \$55000)$$
   $$TC = Min\ : \$3,570,000\ ,\ Avg\ :\ \$8,330,000\ ,\ Max\ :\ \$1,3090,000$$

2. Ignoring the `Sample code number` (SCN), there are
   - A total of 9 attributes for the given
     Δ = Thickness,Cell Size,Cell Shape,Marginal Adhesion, Single Epithelial CS,Bare Nucei,Bland Chromatin,Normal Nucleoli,Mitoses
   - One label value = Class

3. - There are a total of 16 missing values in the column "Bare Nuclei" which are represented with '?'.
   - Corresponding to the missing values n column "Bare Nuclei" there are 16 patients with SCN number as :1057013, 1096800, 1183246, 1184840, 1193683, 1197510, 1241232 , 169356 ,432809, 563649, 606140, 61634, 704168, 33639, 1238464 and 1057067.
   - There are a total of 16 missing records and 8 duplicate data which can be considered as mistakes ine the dataset (Δ).The error rate (ER) then can be calculated as below :

$$\begin{aligned} ER &= \frac{Number\ of\ rows\ with\ errors}{Total\ number\ of\ rows} \\ &= \frac{missing\ values + dulplicate\ rows}{Total\ number\ of\ rows} \\ &= \frac{16+8}{699} = \frac{24}{699} = 3.433\% \end{aligned}$$

- For considering re-examination , lets approach this problem initially with a statistical approach . We can remove the bare nuclei column and still get a PPV value of 94%. This means the removal of the attribute doesn't have much impact on the clustering of data. Hence , we can conclude that re-examination wont have much impact on classification. Also the error rate is quite low : 3.5 % . Couple this with the fact that the column has the lowest entropy or information .Therefore, statistically reexamination is not required.

  However , since this is a high risk calculation and we cant take chances on just statistics , but can use this in aiding which candidates to consider if re-examination is required.Considering the cost of taking a re examination is expensive , anywhere between $1000 to $5000 , we must ideally run the K Means results a number of times and find the false positives which are identified on an average . These records can be recommended a re examinations. Also , before suggesting the re examination we should also run these sets on different algorithms like Naive Bayes and compare the collective results. This would optimize the process of selection of patient for re-examination .

- For finding the missing values , I employed a decision tree to predict the set of values for the column "Bare Nuclei".
  (a) First we have to create an object of decision tree(dtree) class in R . The syntax is as follows : $dtree <- rpart(`BareNuclei` `Thickness`+`CellSize`+`CellShape`+`MarginalAdhesion`+ `SingleEpithelialCS`+`BlandChromatin`+`NormalNucleoli`+`Mitoses`+`Class`, data = cleaned\_data\_breast\_cancer, control = rpart.control(minsplit = 6, cp = 0))$
  (b) The parameter takes three input : The column whose value is to be predicted "Bare Nuclei" , the training dataset set to be used and the control part which specifies the fitting criteria of R.
  (c) Next we predict the values using the following syntax. It takes a decision tree object, the dataset set to predict values for and the types of values. : $tree\_predicton <- predict(dtree, missing_data, type = c("vector"))$
  (d) The predictions are as follows :

SCN	A_1	A_2	A_3	A_4	A_5	$\mathbf{A_6}$	A_7	A_8	A_9	A_{10}
1057013	8	4	5	1	2	**5**	7	3	1	4
1096800	6	6	6	9	6	**10**	7	8	1	2
1183246	1	1	1	1	1	**2**	2	1	1	2
1184840	1	1	3	1	2	**2**	2	1	1	2
1193683	1	1	2	1	3	**2**	1	1	1	2
1197510	5	1	1	1	2	**2**	3	1	1	2
1241232	3	1	4	1	2	**2**	3	1	1	2
169356	3	1	1	1	2	**2**	3	1	1	2
432809	3	1	3	1	2	**2**	2	1	1	2
563649	8	8	8	1	2	**5**	6	10	1	4
606140	1	1	1	1	2	**2**	2	1	1	2
61634	5	4	3	1	2	**2**	2	3	1	2
704168	4	6	5	6	7	**5**	4	9	1	2
733639	3	1	1	1	2	**2**	3	1	1	2
1238464	1	1	1	1	1	**2**	2	1	1	2
1057067	1	1	1	1	1	**2**	1	1	1	2

(a) Only talking about the significance of missing data ,the missing values(16) is not much compared to the total number of cleaned dataset (691) which is approximately just 2% of the total data.Also , the entropy of the column is lowest among all. Hence , just by looking at the numbers we can say that the amount of missing data is not significant.

(b) For the missing values , values were predicted using a decision tree as show in above table . Adding this new values in Δ^* , we can provide this data to the K-means algorithm . Considering this new missing values ($|\Delta| = 691$), we find that the True Positive is 653 and False Positive is 38, giving a PPV of 94.5% by K means algorithm.

The algorithm without considering the the attribute value "Bare Nuclei" and the missing values ($|\Delta| = 675$) gives a True Positive is 665 and False Positive is 26, giving a PPV of 96.23%. Hence , we find that the accuracy is better when the attribute is not considered altogether and very close to the PPV= 97.18% when all attributes are considered with clean data.

Also , when we predict a value and use this value for next prediction , it reduces the accuracy of the Algorithm to predict unknown values, as it has been trained on some results which are not real. Its better to sometimes discard values and train algorithm based on true data rather than some compensated values o get higher accuracy. Also as discussed earlier , the significance of missing data is very less compared to the total rows.

Hence , considering the statistics , we can safely discard the missing values without much lose of information.

4. (a) The following is the code in R for variance function :

```
calVariance <- function(v) {
  m <- 0
  for (i in v)\{
    m <- m + i
  }
  m <- m / length(v)

  variance <- 0
  for (i in v){
    varience <- variance + ( (m - i) ^ 2 )
  }
  print (variance)
  variance <- variance / length(v)
  return (var)
}
```

We find that the maximum variance is for column Bare Nuclei : 13.2145.
The variance table is as shown below :

Column Name	Variance
Thickness (A_1)	7.94546
Cell Size(A_2)	9.333056
Cell Shape (A_3)	8.846736
Marginal Adhesion (A_4)	8.258647
Single Epithelial CS (A_5)	4.870233
Bare Nuclei (A_6)	13.2145
Bland Chromatin (A_7)	6.012673
Normal Nucleoli (A_8)	9.384024
Mitoses (A_9)	3.026612
Class (A_{10})	0.909555

(b) The lowest entropy is of column Bare Nuclei 6.045939. The table representing entropies is as shown below :

Column Name	Entropy
Thickness (A_1)	6.310634
Cell Size(A_2)	6.111481
Cell Shape (A_3)	6.138426
Marginal Adhesion (A_4)	6.105003
Single Epithelial CS (A_5)	6.105003
Bare Nuclei (A_6)	6.045939
Bland Chromatin (A_7)	6.284522
Normal Nucleoli (A_8)	6.06034
Mitoses (A_9)	6.179103
Class (A_{10})	6.455415

(c) The KL distance chart is as below . The KL distance is calculated using the R package entropy and plyr The steps are as below :

i. import packages entropy : library(entropy) and library(plyr)

ii. Calculate the freq using count function and find the probability of each column as a_1&a_2.

iii. the formula to calculate the probability:
$a_1 \leftarrow count(TableName\$ColumnName)\$freq \setminus sum(count(TableName\$ColumnName)\$freq)$

iv. Find the KL distance using formula : $kl\_dist = kl.plugin(a_1, a_2)$

v. Complete the table as shown below :

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
A_1	0	0.338	0.300	0.412	0.612	0.584	0.343	0.550	0.753
A_2	0.332	0	0.009	0.010	1.090	0.106	0.426	0.033	-0.099
A_3	0.287	0.010	0	0.024	0.966	0.167	0.341	0.061	-0.005
A_4	0.389	0.009	0.020	0	1.108	0.112	0.420	0.032	-0.109
A_5	0.830	1.070	0.910	0.986	0	1.501	0.393	1.306	1.483
A_6	0.518	0.090	0.138	0.108	1.418	0	0.722	0.077	-0.384
A_7	0.323	0.460	0.354	0.447	0.445	0.818	0	0.618	0.764
A_8	0.503	0.032	0.055	0.034	1.358	0.080	0.558	0	-0.192
A_9	0.984	0.253	0.289	0.215	1.942	0.268	0.853	0.159	0

The KL distance between attributes of the cancer set.

5. The K-means algorithm is attached . It takes the input of K , the number of columns from dataset and a matrix with binary value (1,0) for selecting that column.The output is partitions in the given number of cluster and the number of runs needed to achieve clustering. The sample output is attached .

   (a) A sample output using all the 9 attributes is :

$$K0 = 438 \ , \ K1 = 237$$
$$TP = 656 \ , \ FP = 19$$

$$PPV = \frac{TP}{TP + FP}$$
$$= \frac{653}{675} = 97.18\%$$

   (b) A sample output using 7 attributes, is :

$$K0 = 446 \ , \ K1 = 229$$
$$TP = 652 \ , \ FP = 23$$

$$PPV = \frac{TP}{TP + FP}$$
$$= \frac{653}{675} = 96.59\%$$

   (c) A sample output using 5 attributes is :

$$K0 = 448 \ , \ K1 = 226$$
$$TP = 648 \ , \ FP = 27$$

$$PPV = \frac{TP}{TP + FP}$$
$$= \frac{653}{675} = 94.96\%$$

   (d) A sample output using 4 attributes is :

$$K0 = 465 \ , \ K1 = 210$$
$$TP = 637 \ , \ FP = 38$$

$$PPV = \frac{TP}{TP + FP}$$
$$= \frac{653}{675} = 94.37\%$$

   We observe that ,certain attribute sets have lesser impact on the over all clustering . Eg Bare Nuclei, Single Epithelial. Also , there are certain correlation among pairs like Cell Size and Cell Shape.Dropping either one of them would result in almost the same results. However , the general trend we see is that by increasing the attributes , we get more accuracy.

6. Following the process explained in the question , we get the following values for each dataset D.All attributes have been considered for evaluating the result.

Train	Test	PPV Result
$\texttt{kmeans}(D^* - \{D_1^*\})$	D_1^*	0.9701
$\texttt{kmeans}(D^* - \{D_2^*\})$	D_2^*	0.9552
$\texttt{kmeans}(D^* - \{D_3^*\})$	D_3^*	0.985
$\texttt{kmeans}(D^* - \{D_4^*\})$	D_4^*	0.9253
$\texttt{kmeans}(D^* - \{D_5^*\})$	D_5^*	0.9552
$\texttt{kmeans}(D^* - \{D_6^*\})$	D_6^*	0.985
$\texttt{kmeans}(D^* - \{D_7^*\})$	D_7^*	0.9552
$\texttt{kmeans}(D^* - \{D_8^*\})$	D_8^*	1
$\texttt{kmeans}(D^* - \{D_9^*\})$	D_9^*	1
$\texttt{kmeans}(D^* - \{D_{10}^*\})$	D_{10}^*	1

The total PPV is then

$$\begin{aligned} PPV(\Delta) &= (1/10)\Sigma_{i=1}^{10}\alpha_i \\ &= \frac{9.731}{10} = 0.9731 = 97.31\% \end{aligned}$$

Thus the results obtained by process of V-fold cross is slightly greater than the previous result of 97.18% , when considering all the attributes of the dataset.