Journal of Computational Information Systems 11: 1 (2015) 1–8
Available at http://www.Jofcis.com

An Improved Imputation Method for Missing Data Based on QENNI ⋆

Zhaoyu Zhang [1,*], Zhibo Chen [2], JianXin Wang [3]

[1] *School of Information Science And Technology, Beijing Forestry University, Beijing 100083, China*
[2] *School of Information Science And Technology, Beijing Forestry University, Beijing 100083, China*
[3] *School of Information Science And Technology, Beijing Forestry University, Beijing 100083, China*

Abstract

Missing data imputation is an important research aspect in data mining. Data quality is a major concern in Machine Learning and other correlated areas such as Knowledge Discovery from Databases (KDD). Many imputation methods of missing data have been designed to resolve the problem. More or less, they have some deficiencies. As the K-Nearst Neighbor Imputation (KNNI) algorithm is often biased in choosing the k mearest neighbors of missing data. A new imputation method is put forward, Quadrant Encapsidated Nearest Neighbor based Imputation method (QENNI). QENNI uses the quadrant nearest neighbors around a missing datum to impute the missing datum. It is not biased in selecting nearest neighbors. Experiments demonstrate that QENNI is much better than the kNNI method in imputed accuracy. But, as the experiment proceeded, we found out the denseness of points in each quadrant and the distance between the two point affect the missing data value badly. So, we improved the QENNI algorithm and put forward Denseness and Distence Weighted Quadrant Encapsidated Nearest Neighbor based Imputation method algorithm (DDWQENNI). The experimental result demonstrates that our DDWQENNI method has a higer imputation accuracy than QENNI.

Keywords: Imputation of missing data; Quadrant; KNNi; QENNI; WQENNI

1 Introduction

Data mining [2] is an interdisciplinary subfied of computer science. It is the computational process of dicovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Data mining is a powerful technology with great potential. Preprocessing as one of the indispensable step of data mining can seriously affect the accuracy

⋆Project supported by the National Nature Science Foundation of China (No. ***).

*Corresponding author.
Email address: china.zhangzhaoyu@gmail.com (Zhaoyu Zhang).

January 2015

of the conclusion. That is why missing data imputation has been an inevitably and challenging research.

Due to its importance in Dataing mining, missing data imputation has received considerable attention during the past decades. A large percentage of studies have been done to develop procedures to deal with missing values. Recently, no matter in which filed of study, $kNNI$ [2] imputation method has been researched and applied widely because of its easy operating, high efficiency an accuracy. It is an excellent imputation algorithm, but thoes choosing nearest neighbors of misssing data is very likely biased in one side. So it's not the best choice to use them for missing data imputation. In addiction, the parameter k is the key factor for the $kNNI$ algorithm. In the experiments, if the k sets a larger value, it brings seriously randomness; if the k sets a smaller value, it will lose large sample size standard of statistics. Before very experiment of $kNNI$, lots of calculation should be token to get the approproate vlaue of k. It makes the algorithm more complex. In response to these prolems, Shichao zhang [2] put forward a new missing data imputation algorithm, quadrant encapsidated nearest neighbor based imputation ($QENNI$). QENNI algorithm imputes the missing data by finding all of the quadrant encapsidated nearest neighbors of the missing data.Exactly to say, it assumes the missing data as the center, the complete data sets are distributed to each quadrant. Because of this feature, it can void the heavily depending on the parameter k of kNNI. Experimental results show QENNI has a higer accuracy than kNNI.

But, according to the analysis of the missing data, we find it also seriously affected by denseness of points in each quadrant and the distance between the missing data and complete data. So, based on QENNI, we take the denseness and distance's weight into account and proposes a new missing data imputation, Denseness and Distence Weighted Encapsidated Nearest Neighbor based Imputation method ($DDWQENNI$). This imputation algorithm overcomes the above-mentioned limitations and has a good performance than QENNI.

The rest of this paper is organized as follows: Section 2 introduces the details of the proposed imputation method and give the corresponding algorithm. Seciont 3 gives the experiments and results. Section 4 discusses the result and drwas a conclusion.

2 DDWQENNI Algorithm

On the basis of the above discussion, in this part, we give the defination and implementation of the DDWQENNI algorithm. The improvement of DDWQENNI algorithm will be pointed out. We also discuss the shortcomings of kNNI and QENNI.

2.1 Algorthm background

Suppose X is an M-dimensional random vector, Y is the dependent variable affected by X. In practice, if a missing data random sample (size is n) can be get, it can be expressed as $(X_i, Y_i, \delta_i), i = 1, 2, \cdots, n$. In which, all of the X_i vector is observable, when Y_i is missing, $\delta_i = 1$, otherwise $\delta_i = 0$. If the data set T contains n data, each data has $m + 1$ attributes (contains m condition attributes and 1 decision attribute), keep: $T_i = (X_{i1}, X_{i2}, \cdots, X_{im}, Y)$ (missing values are generated only in decsion attribute Y). $T = I \bigcap C$, let $r = \sum\limits_{k=1}^{n} \delta_i$, $I = T_1, \cdots, T_r, r \leq n$ are the data sets which decision attrubutes are missing, referred to missing data sets; $C = T_r+_1, \cdots ,T_n$ are the complete data sets.

2.2 *k*NNI algorithm

K-Nearst Neighbor Imputation (kNNI) imputes the missing value by the k nearest neighbors of the missing data. It bases on the theory that the closer the distance, the closer the relation. If a data loses one attrubute, to find out the k nearest neighbors in complete data sets and use the average value of them to impute the missing value.

As previously mentioned, kNNI is praised by the majority of researchers because of its simple operation, low time complexity and high imputation accuracy. But there are still some drawbacks. The k nearest neighbors selected by kNNI algorithm may occur preferences, which makes the filling effect is relatively inefficient. In fact, from the distribution of complete data which surrounds the missing data, the distribution of its nearest neighbors data and overall data may be inconsisten. For instance, as shown in Fig.1, O represents missing data, the other points represents complete data, we use kNNI algorithm to impute the missingd data value. According to the 2.2, we need to find three nearest neighbors and use them to impute the missing data value. So A, B, C are selected. Nevertheless, from Fig.1 we can clearly see that the complete data surrouding the missing data are evenly distributed, but the three selected nearest neighbors bias in the first quadrant. So the three nearest neighbors may not be the best choice and may not be able to get the best results by using them to impute the missing value.

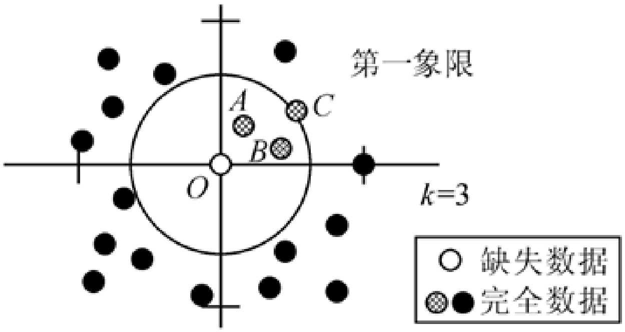

图 1 kNN 算法最近邻点的选择↵

Fig. 1: Nearest neighbors chosen by kNN

To choose the k parameter of kNNI algorithm is diffuclt. Each time, we use kNNI to impute the missing data, we need to repeat the experiment so many times to obtain the value k. Once the value of k occurs deviation, the performance of kNNI will be significantly lower. In conclusion, if the algorithm can eliminate the dependence on the parameter k, it will be the best choice.

2.3 QENNI algorithm

We propose the hypothesis that the complete data which is used to impute missing data must be the nearest neighbors and in the first encirclement of the missingd data. we also need to eliminate the dependence on the parameter k. Let's realize the idea below.

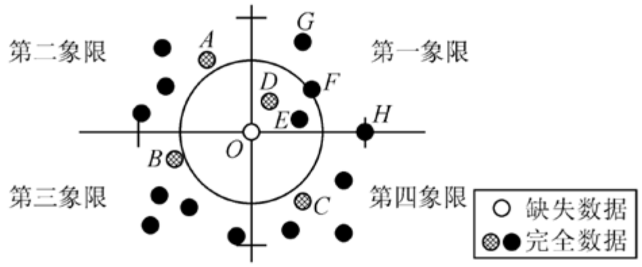

图 2 QENNI 算法最近邻点的选择↵

Fig. 2: Nearest neighbors chosen by QENNI

(1) First, take data $(X_i, X2, \cdots, X_m)$ which contains m condition attributes as a point of m-dimensional space, estabish a coordinate system and the missding data is the center. Through the axis to divide the m-dimensional into 2^m quadrants.

a. When $m = 2$, the condition attributes (X_1, X_2) can be regard as a point of plane. As shown in Fig.2, the axis divides the plane into 2^2 quadrants. If one point is just between the two quadrants, we classify it to one of its nearby quadrant. According to our difinition, everyone of the data set can be located on the only certainty quadrant.

b. Similarly,when $m = 3$, the condition attributes of data is $(X_1, X_2, X3)$. The Formed spatial coordinate system divides the space into 2^3 quadrants and everyone of the data is also in the only identified quadrant.

c. Extended to the general case, when $m = m$, the condition attributes of data is $(X_i, X2, \cdots, X_m)$ and the space is divided into 2^m quadrants.

(2) Based on the dividing of m-dimensional space, the Euclidean distance of each point from its own to the center (which is the missing data) is calculated. To find out the nearest one of each quadrant (if not exists, ignore it) and use decision attributes of them to impute the missing data value. With $m = 2$, for example, as shown in Fig.2. In each quadrant, we select A, B, C, D as the nearest neighbors and use the decision attributes Y of them to impute the missing data value of center O. It also weighted by the distance of each selected point. Obviously, the way of QENNI to choose the nearest neighbors is different from the kNNI algorithm.

$$dist(T_i, T_j) = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2} \tag{1}$$

(3) In order to analyze the effectiveness of the algorithm better, for each missing data $T_i (T_i \in I)$, on the basis of QENNI algorithm, the following definition can be given:

Step 1 The coordinate system centered at T_i divides the space into 2^m quadrants and the complete data set C based on quadrant is divided into 2^m subset $C = \{D_1, D_2, \cdots, D_q, \cdots, D_{2^m}\}$. Each complete data set $D_q (q = 1, 2, \cdots, 2^m)$ is the q quadrant's data of T_i.

Step 2 $\forall T_j \in D_q$, satisfy $Near_q = \arg\min dist_{T_j \in D_q}(T_i, T_j)$, $Near_q$ is nearest neighbor of T_i in q quadrant. As shown in Fig.2, the first quadrant data of T_o is $D_1 = \{T_D, T_E, T_F, T_G, T_H\}$. So $Near_q = T_D$ which is the nearest neighbor of first quadrant.

Step 3 In the q quadrant, take T_i as the center of the sphere (or hypersphere), dist$(Near_q, T_i)$ as the radius to ensure the $Shell_q$ of T_i in q quadrant.

Step 4 All of the $Shell_q(q = 1, 2, \cdots, 2^m)$ and axis constitute the m-dimensional subspace which is Shell of T_i.

Step 5 All the of nearest neighbor $\{Near_1, Near_2, \cdots, Near_{2^m}\}$ of T_i in every quadrant are called the points of Shell.

Nature 1 If $D_q \neq \emptyset$, the Shell of T_i must exist in q quadrant.

Nature 2 $\forall T_j \in D_q$, $\exists dist(T_j, T_i) \geq dist(Near_q, T_i)$.

(4) In summary, the QENNI algorithm can overcome the shortcomings of kNNI in choosing the k nearest neighbors of missing data. The selected complete data is not biased in any side and the Shell of the missing data is smallest. That is to say it does not exist any other complete data on the Shell of the missing data. It is thus clear that QENNI algorithm can find out the most satisfied complete data without any preferences. They can represent the missing data better than kNNI.

2.4 DDWQENNI agloritbm

Even though the QENNI algorithm has a better performace than kNNI, it does not take the denseness into account. As shown in Fig.3, if we just think about the distance of complete data A and D, the A has a higher affect of the result. But it is clear that just the distance can not represents the real affect of the complete data A. If the denseness of complete data which surrounds A is token into accout, it may have a higer weight than D. Based on the above considerations, Our new algorithm takes both the weight of distance and denseness into accout. The fellowing definitions are given.

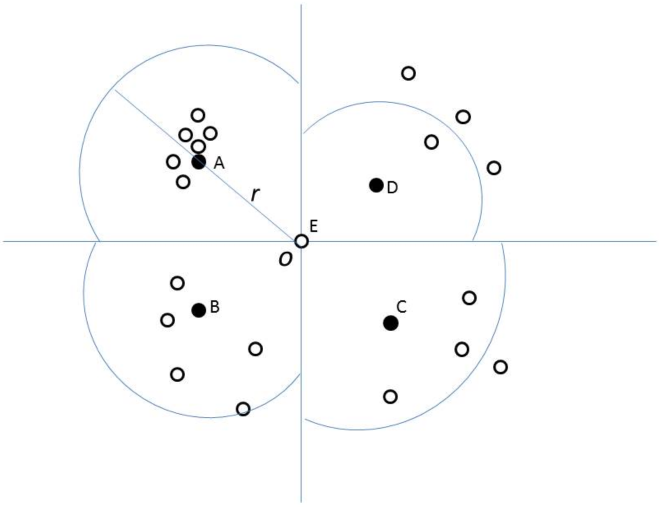

Fig. 3: Nearest neighbors chosen affected by Denseness

(1) Weight of Distance : The different length of distance has differnet size of influence. The more closer, the higher wieght. So we keep weight of each nearest neighbor $Near_i (i = 1, 2, \cdots, 2^m)$ in every quadrant as:

$$W(D)_i = \frac{1}{dist(Near_i, T_i)^2} \tag{2}$$

(2) Weight of Denseness : The denseness represents the number of complete data in per unit volume and it can indicate the wight of the quadrant's denseness. We use Eq. (11) to calculate the volume of n-sphere. The Eq. (13) is used to calculate the denseness which can represents the weight. Each quadrant's number complete data is counted as $N = \{N_1, \cdots, N_i, \cdots, N_{2^m}\}$.

$$V_n(R) = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(\frac{n}{2} + 1)} \tag{3}$$

$$\Gamma(\frac{n}{2} + 1) = \begin{cases} (\frac{n}{2})! & \text{n is Even,} \\ \sqrt{\pi}\frac{n!!}{2^{\frac{n+1}{2}}} & \text{n is Odd.} \end{cases} \tag{4}$$

$$W(\rho)_i = \frac{N_i}{\frac{V_n(R)_i}{2^m}} \tag{5}$$

(3) Core algorithm : The algorithm takes both denseness and distance's weight into consideration and bring in the factor $\beta (0.0 \leq \beta \leq 1.0)$ to decide the percentage of denseness and distance. The detail steps of DDWQENNI are as follows:

a. Calculate the Euclidean distance of missing data set I and complete data set C by Eq. (9).

b. Iterate data set I and C calculate the nearest neighbor, total number of complete data in each quadrant where $R \leq 2 * dist(Near_i, I_i)$ and volume of each hyperspace by Eq. (10, 11, 13).

c. The key about how to figure the complete data for each quadrant is to translate the data vector into a number and use it to mark the quadrant. First, translate the coordinate into a vector consisting of one and zero. if the value of vector greater than zero, translate it into one, otherwise keep it as zero. Than, translate the binary number into decimal.

$$\begin{pmatrix} 2 & -1 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \Rightarrow (1 * 2^2 + 0 * 2^1 + 0 * 2^0) \Rightarrow 4 \tag{6}$$

d. Take the previous results into Eq. (7) to calculate the missing data value (v_i). The equation has token theweight of denseness and distance into account.

$$v_i = \frac{\sum_{i=1}^{n}((1-\beta)W(D)_i + \beta W(\rho)_i) * D_i}{\sum_{i=1}^{n}((1-\beta)W(D)_i + \beta W(\rho)_i} \tag{7}$$

3 Experiments and Results

3.1 Dataset

In order to test our Algorithm, we choose the open dataset Abalone which is from UCI and Delta_ailerons which is from weka. We do the same experiments as QENNI does. The sex attribute is excluded and left the data whoes sex value is M. It has 8 attribute and 1528 records in total. The Diameter attribute is chosen as decision attribute and missing data is generate on it. The Delta_ailerons dataset has 6 attributes and 7129 records in total. The last attribute is chosen as descsion attribute and the missing data is generated on it. The imputation accuracy is used as evaluation index. In general, the Root Mean Square Error (RMSE) is used. e_i is the original value, e_i' is the imputation value and m is the number of missing data. The smaller the RMSE, the higer the imputation accuracy.

$$RMSE = \frac{1}{m}\sum_{1=1}^{m}(e_i - e_i')^2 \tag{8}$$

3.2 Experiments

3.3 Results analysis

4 Conclusion

For improving the efficiency and accuracy of missing data imputaion, DDWQENNI imputation algorithm has been put forward. The method is able to overcome the limitations of kNNI and QENNI. The innovation of our method is to take the denseness of points in each quadrant and distence between the compelte data and the missing data into consideration. So, the imputed data can be more closer to the missing data. The experimental results indicate that DDWQENNI algorithm has a better performance than QENNI. Feature work is to improve the computing speed of WQENNI in hyperspace.

5 Introduction

- **Common** Contributions must be written in English. Each paper should be introduced by a list of keywords and a self-contained abstract of no more than thirty lines without long formulas.
- **Title** Title should be concise but informative. Titles are often used in information-retrieval systems. Avoid abbreviations and formulae where possible.
- **Author** There should be and should only be one corresponding author.
- **Abstract** A concise and factual abstract, of aro und 100 words, is required. The abstract should state briefly the purpose of the research, the principal results and major conclusions. It must be able to stand alone, references should be avoided. Non-standard or uncommon abbreviations should be avoided.

- **Keywords** Three to five keywords are required, using British spelling and avoiding general and plural terms and multiple concepts (avoid, for example, “and”, “of”).

- **Headings** Papers should be divided into numbered sections, subsections and, if necessary, subsubsections (e.g. 3, 3.1, 3.1.1, etc.).

- **Uppercase & Lowercase** Every word within the title of “section” , except empty word, should has its initial capitalized. But for the “subsection” , the only word that should be capitalized is the first one. But note that it is not the case for subsection, see subsection 5.1.

- **Mathematical Symbols** Every mathematical symbol in the text, for example, n, R, x, y etc.

- **Enumerations** Enumerations should be listed in an Item-like environment, e.g. “itemize” “enumerate”.

- **Footnotes** Footnotes should be avoided if possible and as brief as possible, they should be numbered consecutively.

- **Algorithms** If you are presenting an algorithm or listing something with order, make sure you use the “itemize” or “enumerate” environment, treat each step as an “item” and label it as “(n)”, where n is the sequence number of steps. For the sub-items label them as “a.”, “b.” , etc., see section 5.1.

- **Figures** Figures should be numbered consecutively in the order of appearance and citation in the text. Be sure to cite every figure. Handwritten lettering and low-quality computer graphics are not acceptable. EPS electronic files should be sized as they will appear in the journal.

- **Tables** Tables must be numbered and typed on separate pages. The table title, which should be brief, goes above the table. Detailed explanations or table footnotes should be typed directly beneath the table. Note that tables are usually typeset, not scanned (tables cannot be electronically reduced in size).

- **Citations** Citations should coupled with labels. That is, to make a citation , you should label the position first, then use the command “`\ref`”. All citations made in this guide, including equations, tables, figures, etc., follow this rule, you can check the source file to make a clearer understood.

- **References** References must be numbered consecutively in the order of their first citation, as in the following examples: books [2, 6], articles in journals [3], papers in a contributed volume [4, 7], unpublished papers [5].

5.1 Only the first word in the title of “subsection” be capitalized

We place a paradigm for the algorithm here:

(1) The first step.

(2) The second step.

  a. substep1.

  b. substep2.

(3) The last step.

In the ".tex" file it may look like the following:

```
\begin{enumerate}[(1)]
\item The first step.
\item The second step.
  \begin{enumerate}[a.]
  \item substep1.
  \item substep2.
  \end{enumerate}
\item  The last step.
\end{enumerate}
```

You can also use description environment, for example

Step 1 The first step.

Step 2 The second step.

Step 3 The second Step.

In the ".tex" file it may look like the following:

```
\item[Step 1] The first step.
\item[Step 2] The second step.
\item[Step 3] The second Step.
```

Note: Package "enumerate" is needed for this kind of usage of environment of enumerate.

6 Mathematical Notation

6.1 Build-in environments

This document class has provided you some commonly used environments:

- Definition environment
  `\begin{defn}` ······ `\end{defn}`

- Lemma environment
  `\begin{lem}` ······ `\end{lem}`

- Theorem environment
  `\begin{thm} ······ \end{thm}`
- Proof environment
  `\begin{pf*}{Proof} ······ \end{pf*}`
- Corollary environment
  `\begin{col} ······ \end{col}`
- Proposition environment
  `\begin{pro} ······ \end{pro}`

The following examples demonstrate the usage of the above environments.

Definition 1 *A graph G is an ordered pair of disjoint sets (V, E) such that E is a subset of the set of unordered pairs of V.*

Lemma 1 *If $m \geqslant 2n$ then $\epsilon(\overrightarrow{G}; x, y) = 0$.*

Theorem 1 *A graph is bipartite if it does not contain an odd cycle.*

Proof Suppose G is bipartite with vertex classes V_1 and V_2. Let $x_1x_2 \cdots x_l$ be a cycle in G. We may assume that $x_1 \in V_1$. Then $x_2 \in V_2$, $x_3 \in V_1$,and so on: $x_i \in V_1$ if i is odd. Since $x_l \in V_2$, we find that l is even.

Suppose now that G does not contain an odd cycle. Since a graph is bipartite if each component of it is, we may assume that G is connected. Pick a vertex $x \in V(G)$ and put $V_1 = \{y|d(x, y)\text{is odd}\}$, $V_2 = V \backslash V$. There is no edge joining two vertices of the same class V_i since otherwise G would contain an odd cycle. Hence G is bipartite.

Theorem 2 *A graph is a forest if for every pair $\{x, y\}$ of distinct vertices it contains at most one x-y path.*

Proof If $x_1x_2 \cdots x_l$ is a cycle in a graph G then $x_1x_2 \cdots x_l$ and x_1x_l are two x_1-x_l paths in G.

Conversely, let $P_1 = x_0x_1 \cdots x_l$ and $P_2 = x_0y_1y_2 \cdots y_kx_l$be two distinct x_0-x_l paths in a graph G. Let $i+1$ be the minimal index for which $x_{i+1} \neq y_{i+1}$, and let j be the minimal index for which $j \geqslant i$ and y_{j+1} is a vertex of P_1, say $y_{j+1} = x_h$. Then $x_ix_{i+1} \cdots x_ky_jy_{j-1} \cdots y_{i+1}$ is a cycle in G.

Corollary 1 *Every connected graph contains a spanning tree, that is a tree containing every vertex of the graph.*

Proof Take a minimal connected spanning subgraph.

Corollary 2 *A tree of order n has size $n - 1$; a forest of order n with k components has size $n - k$.*

Definition 2 *An oriented graph is a directed graph obtained by orienting the edges, that is by giving the edge ab a direction $\overrightarrow{ab}$ or $\overrightarrow{ba}$. Thus an oriented graph is a directed graph in which at most one of $\overrightarrow{ab}$ and $\overrightarrow{ba}$ occurs.*

Proposition 1 *The set*

$$S_m^\mu(\Delta) = \{f|\ \deg f \leqslant m, f \in S_m^\mu(\Delta)\}$$

is a finite-dimensional linear vector space on k, $m \geqslant 0$.

Lemma 2 *G is Hamiltonian if $C_n(G)$is and G has a Hamilton path if so does $C_{n-1}(G)$.*

Note: If you use the above environments, it will be numbered automatically. If the above environments failed to prove their sufficiency, feel free to define your own theorem-like environments, i.e. `\newtheorem\{Name\}\{Caption\}`.

6.2 Equations

Here are some examples of equations that cover the rules of making a equation with explanations following.

Expressions that are too long or oversized should be separated from the main text, i.e. be surrounded by `$$`$\cdots$`$$`. For example,

$$f(x) = \sum_{k=1}^{\infty} c_k T_{3^k}(x).$$

Never try to number the equation manually. If you want to number a equation, use the corresponding environment, i.e. `Equation` or `Eqnarray` if you want to display mutiple equations with numbers. Eq. (9, 10) and Eq. (11) demonstrate the usage of `Equation` and `Eqnarray` environments respectively.

$$p(x) = a_0 + a_1 + \cdots + a_n x^n. \tag{9}$$

$$[L/M] = \frac{\begin{vmatrix} a_{L-M+1} & a_{L-M+2} & \cdots & a_{L+1} \\ \vdots & \vdots & & \vdots \\ a_L & a_{L+1} & \cdots & a_{L+M} \\ \sum\limits_{j=M}^{L} a_{j-M}X^j & \sum\limits_{j=M-1}^{L} a_{j-M+1}X^j & \cdots & \sum\limits_{j=0}^{L} a_j X^j \end{vmatrix}}{\begin{vmatrix} a_{L-M+1} & a_{L-M+2} & \cdots & a_{L+1} \\ \vdots & \vdots & & \vdots \\ a_L & a_{L+1} & \cdots & a_{L+M} \\ x^M & x^{M-1} & \cdots & 1 \end{vmatrix}}. \tag{10}$$

$$\begin{aligned} K_m(t) &= \frac{1}{(m-1)!} E((x-t)_+^{m-1};\alpha) \\ &= \frac{1}{(m-1)!} \left((\alpha - t)_+^{m-1} - \sum_{k=0}^{n} l_k(\alpha)(x_k - t)_+^{m-1} \right). \end{aligned} \tag{11}$$

Use `displaystyle` to make formulas bigger when necessary.

$$f(z) \approx \frac{1+\dfrac{1}{2}z+z^2+\dfrac{1}{2}z^3}{1-\dfrac{1}{2}z+z^2}. \tag{12}$$

The texts in the equations should not be writing in the mathematical form, you can use `\mbox{#text}` to achieve this, example is given in Eq. (13).

$$f(x) = \begin{cases} 3x^2 & \mbox{when } x \geqslant 0, \\ -3x^2 & \mbox{when } x \leqslant 0. \end{cases} \tag{13}$$

When dealing with well-known functions like min, sin, cos , etc., you should use their normal form in the math environment, i.e. use `\min, \sin, \cos,` $\cdots$ respectively.

$$\arg\min\{\sin x \times \cos(x)\}$$

$$\arg\min\{\sin x \times \cos(x) + f(x) - g(x) + e(x)\},$$

If a sentence is not ended at a equation, the words follows the sentence may not be initial capitalized and intend, see Eq. (14).

Then the unconditional pdf of X is

$$f_X(x) = \int f_{X|\Theta}(x|\theta) f_\Theta(\theta) d\theta, \tag{14}$$

where the integral is taken over all values of θ with positive probability.

7 Table Section

Use “Table” or “Tabular” environment as usual. You may center the table most of the time to beautify your article. You also should name each table. Table [1, 2] are two typical examples of tables.

Table 1: Observation results for LSE

k	1	2	3	4	5	6	7	8
x_k	0	1	2	3	4	5	6	7
y_k	1.4	1.3	1.4	1.1	1.3	1.8	1.6	2.3

8 Figure Section

If you have figures, include them like this:

You can then cite them in your article as following: Fig. 4 shows a process of level set based segmentation.

Table 2: Primitive types in Java

Primitive type	Size	Minimum	Maximum	Wrapper type
boolean	–	–	–	Boolean
char	16-bit	Unicode 0	Unicode $2^{16} - 1$	Character
byte	8-bit	-128	+127	Byte
short	16-bit	-2^{15}	$+2^{15} - 1$	Short
int	32-bit	-2^{31}	$+2^{31} - 1$	Integar
long	64-bit	-2^{63}	$+2^{63} - 1$	Long
float	32-bit	IEEE754	IEEE754	Float
double	64-bit	IEEE754	IEEE754	Double
void	–	–	–	Void

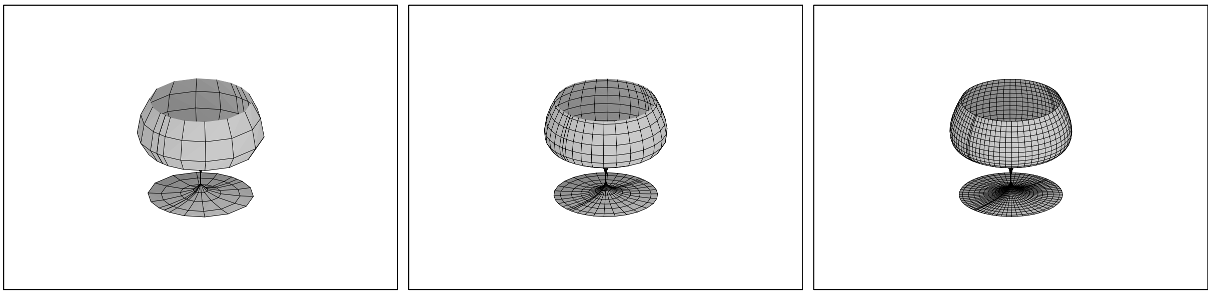

Fig. 4: The control polygon sequences of a cup-like rotation

9 Citing a Reference

You can cite a reference by making use of the command "`\cite`" after you have labelled a bibliography[1]. An illustration of TeX/LaTeXin given in [6]. Please refer to [2, 3, 5, 4] to get a detailed format of references. The citation in the former sentence can be made by using the command "`\cite{NumeApp`, `UncaliEu`, `SpaceDeform`, `Deformation}`", where NumApp, UncaliEu, etc., are user defined labels for references.

Acknowledgement

Acknowledge here.

Appendix

Appendix here.

References

[1] Bibliography, For further detail, please visit our website, http://www.joics.com, 2004

[2] R. H. Wang, *Numerical Approximation*, Higher Education Press, Beijing, 1999

[3] A. Fusiello, Uncalibrated euclidean reconstruction: a review, Image and Vision Computing 18 (2000) 555-563

[4] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, in: Proc. Graphics Interface '95, 1995, pp. 147-154

[5] Y. Sun, Space Deformation with Geometric Constraint, M. S. Thesis, Department of Applied Mathematics, Dalian University of Technology, March 2002

[6] Donald E. Knuth, *The TEXbook*, Addison–Welsey, 1996

[7] E. L. Ortiz, Canonical polynomials in the Lanczos tau-method, in: B. Scaife (Ed.), Studies in Numerical Analysis, Academic Press, New York, 1974, pp. 73-93