Some Operational Computation for Intuitionistic or Pythagorean Fuzzy Set Using C-Programming

Jwngsar Moshahary

1Department of Agricultural Statistics, SCS College of Agriculture,
Assam Agricultural University, India.

Author’s contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/CJAST/2020/v39i2430880

Editor(s):
(1) Samir Kumar Bandyopadhyay, University of Calcutta, India.

Reviewers:
(1) John T. Abraham, Bharata Mata College, India.
(2) Andy Rachman, Institut Teknologi Adhi Tama Surabaya, Indonesia.

Complete Peer review History: http://www.sdiarticle4.com/review-history/59888

ABSTRACT

Intuitionistic or pythagorean fuzzy sets are the best tools to deal with uncertainty or ambiguity to solve diverse disciplines of application problems. It is often difficult to compute union, intersection, and complements when it comes to a large number of members contained in the set, also it is difficult to check whether it is a subset or not. Here, we used the C-programming language to overcome the problems, and then it is found that more effective and realistic results have been obtained.

Keywords: Fuzzy set; intuitionistic fuzzy set; pythagorean fuzzy set; C-programming.

1. INTRODUCTION

Fuzzy sets were introduced by Zadeh [1] in 1965, an extension of the general concept of the set. Zadeh also explained some concepts, namely beautiful, brilliant which do not consist of a usual set to express them in mathematics. For some reason, the concept of fuzziness or ambiguity in use input in practice. In such condition, Zadeh explain the concept of fuzziness by introducing a membership function for a non-empty set X. A fuzzy set A containing a membership function \( f_{A}(x) \) which associates each element \( x \) in \( X \) with a real number \( f_{A}(x) \) in the unit interval \([0,1]\). Since then the concept of “fuzzy set” has been rapid combined with diverse disciplines to solve a
vast number of application problems, which has adequately shown the soundness and the consequence of the fuzzy set theory. Computation of fuzzy set using programming language can be found in. However, the membership function \( f_A(x) \) of a fuzzy set does not totally reflect the vagueness of things, because it cannot convey support, object and uncertainty information in a determination result. After realizing the inadequacy of fuzzy set, In the year 1986 the fuzzy set has been turned to intuitionistic or pythagorean fuzzy set. This will help to find out the larger number of elements containing intuitionistic or pythagorean complement and condition for subset of two intuitionistic or pythagorean fuzzy sets. This will help to find out the larger number of elements containing intuitionistic or pythagorean fuzzy set.

2. PRELIMINARIES

In this section we will give the basic definitions used in the following subsequent sections.

2.1 Definition

Zadeh LA [1] A fuzzy set \( A \) in the non-empty set \( X \) can be defined as a set of ordered pairs and it can be represent mathematically as-

\[ A = \{(x, \mu_A(x)) : x \in X\} \]

Where \( \mu_A(x) \) is membership function of \( x \) in \( X \), such that \( \mu_A(x) : X \rightarrow [0,1] \).

2.2 Definition

Atanassov K [2] An intuitionistic fuzzy set \( A \) in the non-empty set \( X \) can be defined as a set of ordered triplets and it can be represent mathematically as-

\[ A = \{< x, \mu_A(x), \nu_A(x) : x \in X\} \]

Where \( \mu_A(x) : X \rightarrow [0,1] \) is a membership value of \( x \) in \( A \) and \( \nu_A(x) : X \rightarrow [0,1] \) is a non-membership value of \( x \) in \( A \) with \( \mu_A(x) + \nu_A(x) \leq 1 \).

2.3 Definition

Atanassov K [2] Let \( A = \{< x, \mu_A(x), \nu_A(x) : x \in X\} \) and \( B = \{< x, \mu_B(x), \nu_B(x) : x \in X\} \) be two intuitionistic fuzzy sets then

\[
\begin{align*}
(i) & \quad A \cup B = \{< x, \max[\mu_A(x), \mu_B(x)], \min[\nu_A(x), \nu_B(x)] : x \in X\} \\
(ii) & \quad A \cap B = \{< x, \min[\mu_A(x), \mu_B(x)], \max[\nu_A(x), \nu_B(x)] : x \in X\} \\
(iii) & \quad A^c = \{< x, \nu_A(x), \mu_A(x) : x \in X\} \\
(iv) & \quad A \subseteq B \iff \mu_A(x) \leq \mu_B(x) \quad \text{and} \quad \nu_A(x) \geq \nu_B(x) \quad \text{for all} \quad x \in X
\end{align*}
\]

2.4 Definition

Yager RR [4] Let \( X \) be a non-empty set, then the pythagorean fuzzy set (PFS) \( P \) in \( X \) is express as-

\[ P = \{< x, \mu_P(x), \nu_P(x) : x \in X\} \]

Where \( \mu_P : X \rightarrow [0,1] \) signifies the membership value and \( \nu_P : X \rightarrow [0,1] \) signifies non-membership value of the element \( x \in X \) to such set \( P \) respectively. It should fulfill the condition \( \mu_P(x)^2 + \nu_P(x)^2 \leq 1 \).

2.5 Definition

Yager RR [4] Let \( P = \{< x, \mu_P(x), \nu_P(x) : x \in X\} \) and \( Q = \{< x, \mu_Q(x), \nu_Q(x) : x \in X\} \) be two pythagorean fuzzy sets then

\[
(i) \quad P \cup Q = \{<
In this section, we used C-programming language to find the intuitionistic or Pythagorean fuzzy union, intuitionistic or pythagorean intersection, intuitionistic or pythagorean complement and condition for subset of two intuitionistic or pythagorean fuzzy sets.

3. MAIN SECTION

3.1 Program

Source code for C-programming language for the calculation of union of intuitionistic or pythagorean fuzzy sets

```c
#include<stdio.h>
define Max(x,y) (x>y?x:y)
define Min(x,y) (x<y?x:y)
int main()
{
  int r,i,j;
  float A,B,C, D[100][100], E[100][100], F[100][100];
  printf("Enter the number of members of set X: ");
  scanf("%d", &r);
  printf("Enter the elements of intuitionistic or pythagorean fuzzy set A :
  Enter all Membership and non-Membership values respectively");
  for(i=0;i<r;i++)
  {
    for(j=0;j<2;j++)
    {
      scanf("%f",&D[i][j]);
    }
  }
  printf("Enter the elements of intuitionistic or pythagorean fuzzy set B :
  Enter all Membership and non-Membership values respectively");
  for(i=0;i<r;i++)
  {
    for(j=0;j<2;j++)
    {
      scanf("%f",&E[i][j]);
    }
  }
  printf("The intuitionistic or pythagorean fuzzy set A: \n");
  printf("membership\t\t\t\tnon-membership\n");
  for(i=0;i<r;i++)
  {
    printf("\n");
    for(j=0;j<2;j++)
    {
      printf("%f\t\t",D[i][j]);
    }
  }
  printf("\n\n\nThe intuitionistic or pythagorean fuzzy set B: \n");
  printf("membership\t\t\t\tnon-membership\n");
  for(i=0;i<r;i++)
  {
    for(j=0;j<2;j++)
    {
      printf("%f\t\t",E[i][j]);
    }
  }"
printf("\n\n");
for(i=0;i<r;i++)
{
    j=0;
    F[i][j]=Max(D[i][j],E[i][j]);
}
for(i=0;i<r;i++)
{
    j=1;
    F[i][j]=Min(D[i][j],E[i][j]);
}
printf("\n The intuitionistic or pythagorean fuzzy set A union B: \n\n");
printf("membership\tnon-membership\n");
for(i=0;i<r;i++)
{
    printf("\n");
    for(j=0;j<2;j++)
        printf("%f\t\t",F[i][j]);
}
printf("\n\n");
return(0);
}

Example 1: Let \( X = \{x_1,x_2,x_3\} \) then \( A = \{< x_1,0.6,0.2 >,< x_2,0.5,0.4 >,< x_3,0.333,0.666 >\} \) and \( B = \{< x_1,0.777,0.111 >,< x_2,0.888,0.01 >,< x_3,0.999,0.001 >\} \) are intuitionistic fuzzy sets.

**OUTPUT**

![Image of output](image-url)

Hence \( A \cup B = \{< x_1,0.777,0.111 >,< x_2,0.888,0.10 >,< x_3,0.999,0.001 >\} \)

### 3.2 Program

Source code for C programming language for the calculation of intersection of intuitionistic or pythagorean fuzzy sets

```c
#include<stdio.h>
define Max(x,y) (x>y?x:y)
```
#define Min(x,y) (x<y?x:y)
int main()
{
    int r,i,j;
    float A,B,C, D[100][100], E[100][100], F[100][100];
    printf("Enter the number of members of set X: ");
    scanf("%d", &r);
    printf("Enter the elements of intuitionistic or pythagorean fuzzy set A :\n");
    printf("Enter all Membership and non-Membership values respectively");
    for(i=0;i<r;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%f",&D[i][j]);
        }
    }
    printf("The intuitionistic or pythagorean fuzzy set A:
membership
non-membership\n");
    for(i=0;i<r;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
        {
            printf("%f\t\t",D[i][j]);
        }
        printf("\n");
    }
    printf("Enter the elements of intuitionistic or pythagorean fuzzy set B :\n");
    printf("Enter all Membership and non-Membership values respectively");
    for(i=0;i<r;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%f",&E[i][j]);
        }
    }
    printf("The intuitionistic or pythagorean fuzzy set B:
membership
non-membership\n");
    for(i=0;i<r;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
        {
            printf("%f\t\t",E[i][j]);
        }
    }
    for(i=0;i<r;i++)
    {
        j=0;
        F[i][j]=Min(D[i][j],E[i][j]);
    }
    for(i=0;i<r;i++)
    {
        j=1;
        F[i][j]=Max(D[i][j],E[i][j]);
    }
    printf("The intuitionistic or pythagorean fuzzy set A intersection B: \n");
    printf("membership\t\tnon-membership\n");
    for(i=0;i<r;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
        {
            printf("%f\t\t",F[i][j]);
        }
    }
Example 2: From Example 1 We Have

Hence $A \cap B = \{ < x_1, 0.6, 0.2 >, < x_2, 0.5, 0.4 >, < x_3, 0.333, 0.666 > \}$

3.3 Program

Source code for C programming language for the calculation for complement of intuitionistic or pythagorean fuzzy subset $A$.

```c
#include <stdio.h>
int main()
{
    float A[100][100], r;
    int i, j, m, n = 2;
    printf("Enter the number of elements of set X:n");
    scanf("%d", &m);
    printf("Enter the elements of intuitionistic or pythagorean fuzzy set A :
");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
        {
            scanf("%f", &A[i][j]);
        }
    }
    printf("The intuitionistic or pythagorean fuzzy set A:
");
    printf("Membership \n\t\tNon-membership\n");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
            printf(" %f\t\t", A[i][j]);
        printf("\n");
    }
    for (i = 0; i < n; ++i)
        printf("\n");
}
```

OUTPUT

Hence $A \cap B = \{ < x_1, 0.6, 0.2 >, < x_2, 0.5, 0.4 >, < x_3, 0.333, 0.666 > \}$
\[ r = A[i][1]; \\
A[i][1] = A[i][0]; \\
A[i][0] = r; \]

printf("The complement of intuitionistic or pythagorean fuzzy set A is: \n");
printf("Membership \n Non-membership\n");
for (i = 0; i < m; ++i)
{
    for (j = 0; j < n; ++j)
    {
        printf(" %f \t", A[i][j]);
        printf("\n");
    }
    return 0;
}

Example 3: Let \(X = \{x_1, x_2, x_3, x_4\}\), then \(A = \{< x_1, 0.79, 0.55 >, < x_2, 0.37, 0.89 >, < x_3, 0.91, 0.21 >, < x_4, 0.53, 0.76 >\}\) be a pythagorean fuzzy set.

\[
\text{OUTPUT}
\]

Hence \(A^c = \{< x_1, 0.55, 0.79 >, < x_2, 0.89, 0.37 >, < x_3, 0.91, 0.21 >, < x_4, 0.53, 0.77 >\}\)

### 3.4 Program

Source code for C programming language to check intuitionistic or pythagorean fuzzy set \(A\) is subset of intuitionistic or pythagorean fuzzy set \(B\).

```c
#include <stdio.h>
int main()
{
    float a[100][100], b[100][100];
    int i, j, row1, column1=2, flag = 0;
    printf("Number of elements of set X: \n");
    scanf("%d", &row1);
    printf("Enter the elements of intuitionistic or pythagorean fuzzy set A: \n");
    for (i = 0; i < row1; i++)
    {
        for (j = 0; j < column1; j++)
        {
```
```c
scanf("%f", &a[i][j]);
}
}
printf("Enter the elements of intuitionistic or pythagorean fuzzy set B :\n");
for (i = 0; i < row1; i++)
{
    for (j = 0; j < column1; j++)
    {
        scanf("%f", &b[i][j]);
    }
}
printf("n The intuitionistic or pythagorean fuzzy set A: \n");
printf("Membership \t Non-membership\n");
for (i = 0; i < row1; i++)
{
    for (j = 0; j < column1; j++)
    {
        printf("%f \t\t", a[i][j]);
    }
    printf("n");
}

printf("n The intuitionistic or pythagorean fuzzy set B: \n");
printf("Membership \t Non-membership\n");
for (i = 0; i < row1; i++)
{
    for (j = 0; j < column1; j++)
    {
        printf("%f \t\t", b[i][j]);
    }
    printf("n");
}

for (i = 0; i < row1; i++)
{
    if(a[i][0]<=b[i][0] && a[i][1]>=b[i][1])
        continue;
    else
        flag=1;
}
if (flag == 0)
    printf("A is subset of B \n");
else
    printf("A is not subset B \n");
return 0;
```
Example 4: Let $X = \{x_1, x_2\}$ then $A = \{< x_1, 0.76, 0.54 >, < x_2, 0.87, 0.50 >\}$ and $B = \{< x_1, 0.78, 0.47 >, < x_2, 0.90, 0.35 >\}$ are two pythagorean fuzzy sets

```
OUTPUT

(Checking $A \subseteq B$?)

Enter the number of elements of set $X$: 2
Enter the elements of intuitionistic or pythagorean fuzzy set $A$:
0.76 0.54 0.89 0.58
Enter the elements of intuitionistic or pythagorean fuzzy set $B$:
0.78 0.54 0.47 0.35
The intuitionistic or pythagorean fuzzy set $A$:
Membership: 0.760000
Non-membership: 0.540000
The intuitionistic or pythagorean fuzzy set $B$:
Membership: 0.780000
Non-membership: 0.470000
$A$ is subset of $B$
Process returned 0 (0x0) execution time: 54.636 s
Press any key to continue.
```

```
(Checking $B \subseteq A$?)

Enter the number of elements of set $X$: 2
Enter the elements of intuitionistic or pythagorean fuzzy set $A$:
0.76 0.54 0.89 0.58
Enter the elements of intuitionistic or pythagorean fuzzy set $B$:
0.78 0.54 0.47 0.35
The intuitionistic or pythagorean fuzzy set $A$:
Membership: 0.760000
Non-membership: 0.540000
The intuitionistic or pythagorean fuzzy set $B$:
Membership: 0.780000
Non-membership: 0.470000
$A$ is not subset $B$
Process returned 0 (0x0) execution time: 42.351 s
Press any key to continue.
```

4. CONCLUSION

In this paper, we wrote C-programming to compute some basic operations for intuitionistic or pythagorean fuzzy set and given proper examples with verifications. We think this will help to compute the operations easily and effectively for large number of elements contained in intuitionistic or pythagorean fuzzy set which is often time consuming by usual method. Further it can create C-programming for more operations for intuitionistic or pythagorean fuzzy set.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

3. De Supriya K, Biwas R, Roy AR, Some operations on intuitionistic fuzzy sets,


