Έχουμε δύο χρήστες. Χρειαζόμαστε μία μεταβλητή x12 που θα μας λέει αν ο χρήστης 1 θα ταιριάξει με τον χρήστη 2. Γενικά, αν έχουμε n χρήστες θα χρειαστούμε τις μεταβλητές

xij

για κάθε 1 <= i < j <= n

Π.χ. αν έχουμε 4 χρήστες θα χρειαστούμε τις μεταβλητές x1,2, x1,3, x1,4, x2,3, x2,4, x3,4

Στο πρόβλημα με τους 2 χρήστες πρέπει να ισχύουν οι εξής περιορισμοί:

Integer x\_1\_2

0 <= x\_1\_2 <= 1

x\_1\_2 = 1

O περιορισμός 0 <= x\_1\_2 <= 1 σε συνδυασμό με τον «Integer x\_1\_2» μας λέει ότι το x\_1\_2 θα είναι είτε 0 (ο χρήστης 1 δεν μπαίνει μαζί με τον χρήστη 2) είτε 1 (ο χρήστης 1 μπαίνει μαζί με τον χρήστη 2).

O τελευταίος περιορισμός προκύπτει από το ότι ο χρήστης 1 πρέπει να ταιριάξει μόνο με έναν άλλο χρήστη και έτσι το άθροισμα όλων των μεταβλητών x\_ που έχουν σε κάποιον από τους δείκτες την τιμή «1» πρέπει να είναι ίσο με 1.

*Σημείωση: αντί για μία μεταβλητή xij όπου ι < j μπορούν να χρησιμοποιηθούν δύο μεταβλητές xij και xji* αλλά επειδή προφανώς αν ο χρήστης i τοποθετηθεί μαζί με τον j, τότε και ο j θα τοποθετηθεί μαζί με τον i,θα πρέπει *xij = xji*. Στα επόμενα έχει επιλεγεί να χρησιμοποιείται μία μόνο μεταβλητή.

Αντίστοιχα, αν είχαμε 4 χρήστες, θα έπρεπε να έχουμε:

Integer x\_1\_2, x\_1\_3, x\_1\_4, x\_2\_3, x\_2\_4, x\_3\_4

0 <= x\_1\_2 <= 1

0 <= x\_1\_3 <= 1

0 <= x\_1\_4 <= 1

0 <= x\_2\_3 <= 1

0 <= x\_2\_4 <= 1

0 <= x\_3\_4 <= 1

x\_1\_2 + x\_1\_3 + x\_1\_4 = 1

x\_1\_2 + x\_2\_3 + x\_2\_4 = 1

x\_1\_3 + x\_2\_3 + x\_3\_4 = 1

Αυτά για τους περιορισμούς. Συνολικά, το πλήθος των μεταβλητών θα είναι [n \* (n-1) / 2] και κάθε μία από αυτές θα έχει έναν περιορισμό ότι πρέπει να είναι ακέραιη και ότι 0 <= x\_i\_j <= 1. Επίσης θα υπάρχουν n-1 περιορισμοί αθροίσματος.

Αυτά για τους περιορισμούς. Πάμε τώρα στο utility function.

Προσαρμόζοντας τον τύπο που είχαμε πει για το utility function στις αρχικές επικοινωνίες θα έχουμε ότι από την προοπτική του χρήστη I το όφελος της συνεργασίας του με τον χρήστη j δίνεται ως εξής:

Από το γεγονός ότι οι δύο χρήστες έχουν βαθμολογήσει ο ένας τον άλλο στο παρελθόν καταλαβαίνουμε ότι έχουν ξανασυνεργαστεί. Από το userCollaborationIntentions καταλαβαίνουμε ότι ο χρήστης 1 είναι ουδέτερος ως προς το να ξανασυνεργαστεί με τον χρήστη 2 ενώ ο χρήστης 2 έχει πει ότι θέλει να συνεργαστεί με τον χρήστη 1. Άρα:

Για τον χρήστη #1, το utility w\_1\_2 για τη συνεργασία του με τον χρήστη #2 είναι

w\_1\_2 = (3.8 + 4) / 10 – 0.5 = 0.78 – 0.5 = 0.28

Για τον χρήστη #2, το utility w\_2\_1 για τη συνεργασία του με τον χρήστη #1 είναι

w\_2\_1 = 1 + (4.9 + 2) / 10 = 1 + 0.69= 1.69

Και το συνολικό objective function είναι

x\_1\_2 \* w\_1\_2 + x\_1\_2 \* w\_2\_1

Αυτό πρέπει να γίνει maximize.

Αν είχαμε 4 χρήστες, το συνολικό objective function θα ήταν

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | + | x\_1\_2 \* w\_1\_2 | + | x\_1\_3 \* w\_1\_3 | + | x\_1\_4 \* w\_1\_4 | + |
| x\_1\_2 \* w\_2\_1 |  |  | + | x\_2\_3 \* w\_2\_3 | + | x\_2\_4 \* w\_2\_4 | + |
| x\_1\_3 \* w\_3\_1 | + | x\_2\_3 \* w\_3\_2 | + |  |  | x\_3\_4 \* w\_3\_4 | + |
| x\_1\_4 \* w\_4\_1 | + | x\_2\_4 \* w\_4\_2 | + | x\_3\_4 \* w\_4\_3 |  |  |  |