STABLE MARRIAGE ALGORITHM USING SKYLINES

Sudhir(12734), Rohit Anurag(12585), Rahul Nayak(12538)

November 20, 2015

Indian Institute of Technology, Kanpur

ABSTRACT

Job of a good matchmaker can be considered one of the toughest job. His role is to make an arrangement which is stable and does not lead to a breakup in the future. With n number of female clients and male clients our role would be to assign to each of them a good partner according to their preference. Earlier David Gale and Lloyd Shapley tried to solve this problem but his algorithm on application can prove to be a bit more time and space complex. We in our project tried to solve this issue by introducing skylines in it. It not only makes the algorithm more efficient but also gives it a chance to be more practical and useful for people.

PROBLEM STATEMENT

Given two datasets A and B where A and B contains attributes of there qualities and some requirement attributes for other data's quality (A contains for B's qualities and B for A's qualities), our goal is to calculate the preferences for each $b \in B, \forall a \in A$,and for each $a \in A, \forall b \in B$ and get a stable matching between set A and B.

STABLE MATCHING

Suppose $a_1 \in A$,and $b_1 \in B$ and $a_1 \diamond b_1$ where '$\diamond$' means items are matched. This match is said to be stable if

$$\nexists\, a_2 \diamond b_2 : a_2 \succ_{b_1} a_1,\ \ b_2 \succ_{a_2} b_1 \tag{1}$$

where $\succ_x$ represents preference for x over two elements.

PREVIOUS WORK

GALE-SHAPLEY

Input: Two sets M and W and given there preferences $\forall m \in M$ over $\forall w \in W$, and $\forall w \in W$ over $\forall m \in M$
Output: Return Stable matched set.

GALE-SHAPLEY...

Initialize all m $\in$ M and w $\in$ W to free

GALE-SHAPLEY...

```
Initialize all $m \in M$ and $w \in W$ to free
while $\exists$ free man $m$ do
```

GALE-SHAPLEY...

```
Initialize all $m \in M$ and $w \in W$ to free
while $\exists$ free man m do
    w = highest ranked woman to whom  m has not proposed
```

GALE-SHAPLEY...

```
Initialize all m ∈ M and w ∈ W to free
while ∃ free man m do
    w = highest ranked woman to whom  m has not proposed
    if w is free then
        (m,w) become engaged
    else
        some pair (m’,w) already exists
```

GALE-SHAPLEY...

```
Initialize all $m \in M$ and $w \in W$ to free
while $\exists$ free man m do
    w = highest ranked woman to whom  m has not proposed
    if w is free then
        (m,w) become engaged
    else
        some pair (m’,w) already exists
        if w prefers m to m’ then
            (m,w) becomes engaged
            m’ becomes free
        else
```

GALE-SHAPLEY...

```
Initialize all m $\in$ M and w $\in$ W to free
while $\exists$ free man m do
    w = highest ranked woman to whom  m has not proposed
    if w is free then
        (m,w) become engaged
    else
        some pair (m',w) already exists
        if w prefers m to m' then
            (m,w) becomes engaged
            m' becomes free
        else
            (m',w) remain engaged
        end
    end
```

GALE-SHAPLEY...

```
Initialize all m ∈ M and w ∈ W to free
while ∃ free man m do
    w = highest ranked woman to whom  m has not proposed
    if w is free then
        (m,w) become engaged
    else
        some pair (m’,w) already exists
        if w prefers m to m’ then
            (m,w) becomes engaged
            m’ becomes free
        else
            (m’,w) remain engaged
        end
    end
    return (matched,unmatched)
end
```

Algorithm 7: Gale-Shapley Algorithm

OUR APPROACH

REQUIREMENT OR PREFERENCE FUNCTION

A preference function is a linear function in which coefficients are values from one data set (say A) and variable values are taken from another set's (B's) attributes to calculate the preference. Formally, Let A and B are two sets. A has some quality attributes $\{ a_{Q_1}, \ldots, a_{Q_q} \}$ and some requirement attributes $\{ a_{R_1}, \ldots, a_{R_r} \} \forall a \in A$ and B also has the same definition. Now requirement function f(a) for $a \in A$ and f(b) for $b \in B$ is defined as:

REQUIREMENT OR PREFERENCE FUNCTION

A preference function is a linear function in which coefficients are values from one data set (say A) and variable values are taken from another set's (B's) attributes to calculate the preference. Formally, Let A and B are two sets. A has some quality attributes $\{ a_{Q_1}, \ldots, a_{Q_q} \}$ and some requirement attributes $\{ a_{R_1}, \ldots, a_{R_r} \}$ $\forall a \in A$ and B also has the same definition. Now requirement function f(a) for $a \in A$ and f(b) for $b \in B$ is defined as:

$$f(a) = a_{R_1} * b_{Q_1} + a_{R_2} * b_{Q_2} + \cdots + a_{R_r} * b_{Q_r} \tag{2}$$

REQUIREMENT OR PREFERENCE FUNCTION

A preference function is a linear function in which coefficients are values from one data set (say A) and variable values are taken from another set's (B's) attributes to calculate the preference. Formally, Let A and B are two sets. A has some quality attributes $\{ a_{Q_1}, \ldots, a_{Q_q} \}$ and some requirement attributes $\{ a_{R_1}, \ldots, a_{R_r} \}$ $\forall a \in A$ and B also has the same definition. Now requirement function f(a) for $a \in A$ and f(b) for $b \in B$ is defined as:

$$f(a) = a_{R_1} * b_{Q_1} + a_{R_2} * b_{Q_2} + \cdots + a_{R_r} * b_{Q_r} \tag{2}$$

$$f(b) = b_{R_1} * a_{Q_1} + b_{R_2} * a_{Q_2} + \cdots + b_{R_q} * a_{Q_q} \tag{3}$$

REQUIREMENT OR PREFERENCE FUNCTION

A preference function is a linear function in which coefficients are values from one data set (say A) and variable values are taken from another set's (B's) attributes to calculate the preference. Formally, Let A and B are two sets. A has some quality attributes $\{ a_{Q_1}, \dots, a_{Q_q} \}$ and some requirement attributes $\{ a_{R_1}, \dots, a_{R_r} \} \forall a \in A$ and B also has the same definition. Now requirement function f(a) for $a \in A$ and f(b) for $b \in B$ is defined as:

$$f(a) = a_{R_1} * b_{Q_1} + a_{R_2} * b_{Q_2} + \cdots + a_{R_r} * b_{Q_r} \tag{2}$$

$$f(b) = b_{R_1} * a_{Q_1} + b_{R_2} * a_{Q_2} + \cdots + b_{R_q} * a_{Q_q} \tag{3}$$

To normalize the preference function we insist that $0 \leq a_{R_i} \leq 1$ and $0 \leq b_{R_i} \leq 1 \forall i \in \{1, \dots, r\}$. Note that number of dimensions for requirement attributes for $a \in A$ should be equal to quality dimensions in set B and same applies for $b \in B$'s requirement attributes.

MODIFICATION IN GALE-SHAPLEY ALGORITHM (M-GSA)

Input: Set A and B with there requirement and quality attributes.

Output: Return matched and unmatched pairs.

MODIFICATION IN GALE-SHAPLEY ALGORITHM (M-GSA)

Input: Set A and B with there requirement and quality attributes.

Output: Return matched and unmatched pairs.

```
Calculate preference $\forall a \in A$ and $\forall b \in B$ by their requirement function
Initialize all $a \in A$ and $b \in B$ to free
```

MODIFICATION IN GALE-SHAPLEY ALGORITHM (M-GSA)

Input: Set A and B with there requirement and quality attributes.

Output: Return matched and unmatched pairs.

```
Calculate preference $\forall a \in A$ and $\forall b \in B$ by their requirement function
Initialize all $a \in A$ and $b \in B$ to free
if sizeOf(A) < sizeOf(B) then
    Interchange A and B variables.
end
```

MODIFICATION IN GALE-SHAPLEY ALGORITHM (M-GSA)

Input: Set A and B with there requirement and quality attributes.

Output: Return matched and unmatched pairs.

```
Calculate preference $\forall a \in A$ and $\forall b \in B$ by their requirement function
Initialize all $a \in A$ and $b \in B$ to free
if sizeOf(A) < sizeOf(B) then
    Interchange A and B variables.
end
while $\exists$ free item $a \in A$ do
    b = highest ranked item $\in B$ to whom  a has not tried a match
    if b is free then
        (a,b) become matched
    else
        ⋮
    end
end
return matched(A,B), unmatched(B)
```

Algorithm 11: Modified Gale-Shapley Algorithm

stable marriage skyline algorithm (sms)

Input: Set A and B with there requirement and quality attributes.

Output: Return stable matched pairs.

R_A = RTree(A)
R_B = RTree(B)

stable marriage skyline algorithm (sms)

Input: Set A and B with there requirement and quality attributes.

Output: Return stable matched pairs.

```
$R_A$ = RTree(A)
$R_B$ = RTree(B)
while $R_A$ not empty and $R_B$ not empty do
    // use Branch and Bound Algorithm to Compute Skyline
    Sky(A) = BBS($R_A$)
    Sky(B) = BBS($R_B$)
```

STABLE MARRIAGE SKYLINE ALGORITHM (SMS)

Input: Set A and B with there requirement and quality attributes.

Output: Return stable matched pairs.

```
$R_A$ = RTree(A)
$R_B$ = RTree(B)
while $R_A$ not empty and $R_B$ not empty do
    // use Branch and Bound Algorithm to Compute Skyline
    Sky(A) = BBS($R_A$)
    Sky(B) = BBS($R_B$)
    // use our modified Gale-Shapley algorithm to get matched and
    unmatched pair
    (matchedA,matchedB), unmatched = M-GSA(Sky(A),Sky(B))
```

STABLE MARRIAGE SKYLINE ALGORITHM (SMS)

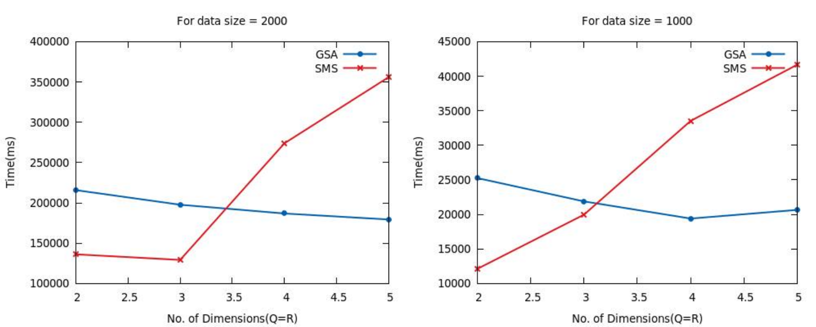
Input: Set A and B with there requirement and quality attributes.

Output: Return stable matched pairs.

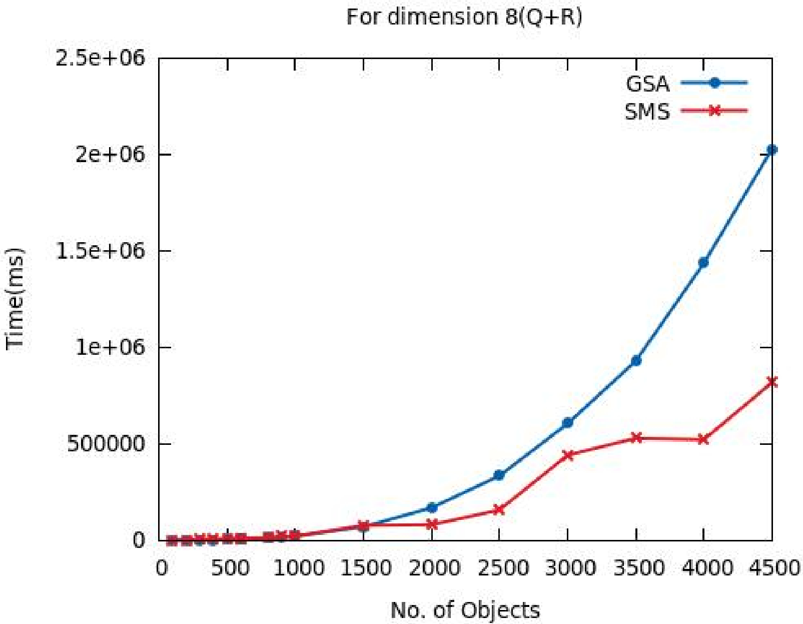
```
$R_A$ = RTree(A)
$R_B$ = RTree(B)
while $R_A$ not empty and $R_B$ not empty do
    // use Branch and Bound Algorithm to Compute Skyline
    Sky(A) = BBS($R_A$)
    Sky(B) = BBS($R_B$)
    // use our modified Gale-Shapley algorithm to get matched and
    unmatched pair
    (matchedA,matchedB), unmatched = M-GSA(Sky(A),Sky(B))
    for each a in matchedA do
        delete a from $R_A$
    end
    for each b in matchedB do
        delete b from $R_B$
    end
end
```

RESULTS & COMPARISONS

FOR SAME DATA SIZE



FOR SAME DIMENSIONS



REFERENCES

- Gale-Shapley Algorithm: D.GALE L.S.SHAPLEY
  paper link
- Fair Assignment based on multiple prefrence queries: Leong Hou U,Nikos Mamoulis,Kyriakos Mouratidis
  paper link
- An Optimal and Progressive Algorithm for Skyline Queries: Dimitris Papadias, Yufei Tao, Greg Fu, Bernhard Seeger
  paper link
- R-TREES. A DYNAMIC INDEX STRUCTURE FOR SPATIAL SEARCHING: Antomn Guttman paper link
- https://en.wikipedia.org/wiki/R-tree

Thank you