Detailed Implementation of the Gale-Shapley Algorithm

Supplemental notes for the lecture on September 17, 2014

This note has been typed up in a bit of hurry so there might be typos in here. If you find one, please post it on piazza. –Atri

This note presents the full detailed implementation of the Gale-Shapley algorithm with all the data structures explicitly initialized and used in the algorithm statement. For a higher level overview of the Gale-Shapley algorithm, please see the lecture slides. Also we present the instantiations for two of 2D array data structures to better illustrate how the algorithm works.

1 Detailed Implementation of the Gale-Shapley Algorithm

Algorithm 1 is the detailed implementation of the Gale-Shapley algorithm. Few reminders/comments:

1. Recall that given an integer $n \geq 1$, we define $[n]$ to be the set $\{1, \ldots, n\}$.
2. We think of the set of men as $[n]$ and the set of women also as $[n]$. In this case a man $m \in [n]$ refers to the ith man (for some fixed ordering of the men) and $w \in [n]$ refers to the ith women (for some fixed ordering of the women).
3. $\mathsf{ManPref}[m, j]$ is the identity of the jth ranked woman in m's preference list and $\mathsf{WomanPref}[w, j]$ is the identity of the jth ranked man in w's preference list.
4. FreeWomenList is a linked list of free women. FreeWomenList.Delete() returns the entry in the front of the list (and deletes it from the list) and returns NULL if the list is empty while FreeWomenList.Insert(w) inserts w at the front of the list.
5. Next is an array of length n, where $\mathsf{Next}[w]$ is the *rank* of the best unproposed man for w. The identity of the best unproposed man can be accessed from $\mathsf{WomanPref}[w, \mathsf{Next}[w]]$.
6. Current is an array of length n such that $\mathsf{Current}[m]$ is the identity of the woman m is currently engaged to and is -1 otherwise.
7. Rank is an $n \times n$ 2D array such that $\mathsf{Rank}[m, w]$ is the *rank* of w in m's preference list.

2 A worked out example for $n = 3$

The 2D arrays ManPref, WomanPref and Rank seems to give some of you a bit of grief, so let me just give a simple example for $n = 3$ and then show what these 2D arrays look like for this instance.

Let us start with an instance where we have *not* mapped the set of men and women to $[n]$. In particular, let us assume that

$$M = \{m_1, m_2, m_3\} \text{ and } W = \{w_1, w_2, w_3\}.$$

Algorithm 1 Gale-Shapley Algorithm

INPUT: $n \times n$ 2D arrays ManPref and WomanPref
OUTPUT: A stable matching S

```
FreeWomenList ← [n]                  ▷ The linked list of free women is initialized with all women
FOR every i = 1...n DO
    Next[i] ← 1                      ▷ The rank of the best unproposed man for every woman is initialized to 1
    Current[i] ← −1                  ▷ All men are initially not engaged
FOR every m = 1...n DO               ▷ Initializing the Rank matrix
    FOR every j = 1...n DO
        Rank[m, ManPref[m, j]] ← j

w ← FreeWomenList.Delete()           ▷ w is a free woman
WHILE w ≠ NULL and Next[w] ≤ n DO    ▷ Check if there is an unproposed man for free w
    m ← WomanPref[w, Next[w]]        ▷ m is the best unproposed man for w
    w′ ← Current[m]
    IF w′ = −1 THEN                  ▷ m is free
        Current[m] ← w               ▷ (m, w) get engaged
    ELSE
        IF Rank[m, w′] < Rank[m, w] THEN   ▷ m prefers w′ to w
            FreeWomenList.Insert(w)  ▷ w remains a free woman
        ELSE                         ▷ m prefers w to w′
            Current[m] ← w           ▷ (m, w) are engaged
            FreeWomenList.Insert(w′) ▷ w′ is now free
    Next[w] ← Next[w] + 1            ▷ Updating the next best unproposed man for w
    w ← FreeWomenList.Delete()

S ← ∅
FOR every m = 1...n DO
    S ← S ∪ {(m, Current[m])}        ▷ At the end of the algorithm m is engaged to Current[m]
RETURN S
```

Further, here are the preference lists of men

$$L_{m_1}:\ w_1 > \ w_2 > \ w_3$$
$$L_{m_2}:\ w_3 > \ w_1 > \ w_2$$
$$L_{m_3}:\ w_2 > \ w_3 > \ w_1,$$

and here are the preference lists for women:

$$L_{w_1}:\ m_1 > \ m_2 > \ m_3$$
$$L_{w_2}:\ m_1 > \ m_2 > \ m_3$$
$$L_{w_3}:\ m_1 > \ m_2 > \ m_3.$$

Now we move from $M = \{m_1, m_2, m_3\}$ to $M = [3]$ where $i \in [3]$ refers to the man m_i. Similarly we think of $W = [3]$, where again $i \in [3]$ will refer to the woman w_i. With this convention and the definitions from earlier, this is how the matrix for $\mathsf{ManPref}$ will look like

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix},$$

where the (m, j) entry above (for $m, j \in [3]$, where m is the row index and j is the column index) is the identity of the jth ranked woman in m's preference list. $\mathsf{WomanPref}$ will look like this

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix},$$

where the (w, j) entry above $(w, j \in [3]$, where w is the row index and j is the column index) is the identity if the jth ranked man in w's preference list.

Below is what Rank looks like (remember that Rank only depends on $\mathsf{ManPref}$):

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix},$$

where the (m, w) entry (for $m, w \in [3]$, where m is the row index and w is the column index) denotes the rank of w in m's preference list.