CS 217: Artificial Intelligence and Machine Learning **Jan-Apr 2024**

Lecture 23: Voting Rules and Stable Matching

Lecturer: Swaprava Nath *Scribe(s): SG45 & SG46*

23.1 Voting rules and Desirable properties

In the last lecture, we had seen the following voting rules:

1) **Plurality**
2) **Borda's rule**
3) **Single Transferable Vote (STV)**
4) **Copeland Voting**

We had also seen a desirable property for voting rules - Condorcet Consistency.

Condorcet Consistency – If a candidate beats all other candidates in pairwise election, then that candidate should win. A voting rule, which obeys this, is said to be condorcet consistent.

Now, we shall see two other desirable properties for voting rules - **unanimity** and **non-manipulability**.

23.1.1 Unanimity

A voting rule f is **unanimous** if for every profile P s.t $P_1(1) = P_2(1) = P_3(1) = \cdots = P_n(1) = a$ (say), it holds that $f(P) = a$.

[Recall the definition of a voting rule f, from the set of profiles to the set of alternatives (candidates) $A = \{a_1, a_2, \cdots, a_m\}$. A profile is of the form $P = (P_1, P_2, \cdots, P_n)$ where n = number of agents (voters) and each voter's P_i is a strict preference order among the elements of A, which is represented as,

$$P_i = \left[\begin{array}{c|c} a & P_i(1) \\ b & P_i(2) \\ c & P_i(3) \\ . & . \\ . & . \\ . & P_i(m) \end{array}\right]$$

$$f(P) = f(P_1, P_2, P_3, \ldots, P_n) \in A$$

]
All the voting rules discussed before - Plurality, Borda's rule, STV, Copeland Voting - are unanimous, since if the top candidates of all the voters are the same, these voting rules make that candidate the winner. Below is the definition, just in case.

1. **Plurality:** In plurality, a candidate wins when they have more votes than any other candidate. If all the agents choose a specific candidate as their most favorable candidate, that candidate has ensured a victory with 100% of the votes according to the plurality voting rule. Therefore, plurality is unanimous.
2. **Borda's rule**: In Borda's rule, each vote contributes to the candidate's score by an amount weighted by how highly they were preferred. Suppose all the agents choose a specific candidate as their most favorable candidate. In that case, that candidate achieves the maximum attainable score according to Borda's voting rule and is ensured a victory as more than one candidate can't attain the maximum attainable score.
3. **STV**: In the STV voting rule, the candidate with the minimum plurality score is eliminated in each round. If all the agents choose a specific candidate as their most favorable candidate, then that candidate wins all the rounds as the plurality voting rule is unanimous. Therefore, it is pretty evident that the STV rule is also unanimous.
4. **Copeland:** Copeland works by performing pairwise face-offs among candidates where a candidate gains a point on winning more votes and 0.5 points in case of a draw. If all the agents choose a specific candidate as their most favorable candidate, then that candidate achieves a score of m-1(where m is the number of candidates contesting the election). This score is strictly greater than the score of all the other candidates, as they can have a maximum score of m-2. So, we can conclude that Copeland's voting rule is Condorcet consistent.

23.1.2 Non-Manipulability

Q:- When is a voting rule manipulable?
A voting rule is manipulable when $f(P_i, P_{-i}) = b$, but upon manipulating the preference order from P_i to P_i', it becomes $f(P_i', P_{-i}) = a$, where the i-th voter prefers a more than b, in his actual preference order P_i.

$$P_i = \begin{bmatrix} . \\ a \\ . \\ b \\ . \\ . \end{bmatrix} \quad P_i' = \begin{bmatrix} a \\ . \\ . \\ . \\ b \\ . \end{bmatrix}$$

A voting rule f is **manipulable** if $\exists\ i \in N$, profile $P = (P_i, P_{-i})$ such that $f(P_i', P_{-i})$ is more preferred by i under P_i than $f(P_i, P_{-i})$ for some P_i'. A voting rule f is **non-manipulable** if f doesn't satisfy the above condition.

Now, we shall see examples of profiles P that show that the voting rules seen until now - Plurality, Borda's rule, STV, Copeland Voting - are manipulable.

1) Plurality rule: Consider the following example.

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ a & a & b & b & c & d \\ b & b & a & a & b & a \\ c & d & c & c & a & b \\ d & c & d & d & d & c \end{bmatrix}$$

By the plurality rule, there's a tie between a and b as top-order choices, each with 2 votes. Now, if the tie is broken in favour of a so that a becomes the winner, then voter 5 can manipulate his voting preference by changing P_5 to P_5' as shown below. Then b becomes the clear winner (3 votes).

$$P_5 = \begin{bmatrix} c \\ b \\ a \\ d \end{bmatrix} \quad \rightarrow P_5' = \begin{bmatrix} b \\ c \\ a \\ d \end{bmatrix}$$

Similarly if the tie is broken in favour of b so that b is the winner, then the voter 6 can manipulate by putting a as his first preference so that a becomes the winner.
Therefore, **Plurality is manipulable.**

2) Bordas's rule: Consider the previous example.

This rule is basically assigning some weights to the preference order,i.e 3,2,1,0 to the corresponding rows of the matrix. By calculating the Borda scores we get $a = 13$, $b = 13$, $c = 6$, $d = 4$ (a tie). If the tie is broken in the favour of a so that a is the winner $[f_{Borda}(P) = a]$, then P_5 manipulates to P_5' in order to make b as the winner. Similarly, if the tie is broken in favour of b, then voter 6 can manipulate to make the a the winner.
Therefore, **Borda's rule is manipulable.**

3) Single Transferable Voting: Consider the following example.

$$P = \begin{bmatrix} P_1 & P_2 & P_3 \\ a & b & c \\ b & c & a \\ c & a & b \end{bmatrix}$$

Here, the plurality scores (number of voters for which the candidate is the top-order choice) are equal to 1 for all the candidates a, b, c. Without loss of generality, if tie is broken against c, then c will get eliminated as one with the minimum plurality score. Out of the remaining candidates, a will have a plurality score of 2 and b will have 1, and hence b will get eliminated next, making a the winner.

$$P_2 = \begin{bmatrix} b \\ c \\ a \end{bmatrix} \quad \rightarrow P_2' = \begin{bmatrix} c \\ b \\ a \end{bmatrix}$$

Then, voter 2 can manipulate his preference order as shown, so that $f(P_2', P_{-2}) = c$ since c has a higher preference than a for him and c becomes the winner with a plurality score of 2 (highest), a, b with plurality scores 1,0 respectively, since b gets eliminated first and then a gets eliminated next for having minimum plurality score.
Therefore, **STV rule is manipulable**.

4) Copeland rule: Consider the same example used above.

Here the Copeland scores (number of pairwise wins) for each of the three candidates a,b,c are equal to 1 (a tie). Without loss of generality, if the tie is broken in favour of a so that a is the Copeland winner, i.e., $f(P) = a$.

$$P_2 = \begin{bmatrix} b \\ c \\ a \end{bmatrix} \quad \rightarrow P_2' = \begin{bmatrix} c \\ b \\ a \end{bmatrix}$$

Then voter 2 can manipulate his preference order so that $f(P_2', P_{-2}) = c$ since c has a higher preference than a for him and c becomes the winner with Copeland score of 2 (highest), a, b with Copeland scores 1,0 respectively.
Therefore, **Copeland rule is manipulable.**

23.1.3 Dictatorial Voting Rule

In a dictatorial voting rule f, $\exists$ d $\in$ N s.t $f(P_d, P_{-d}) = P_d(1)$
A dictatorial voting rule is **unanimous** and **non-manipulable**.

Gibbard-Satterthwaite theorem:
If voters can have all the possible strict preferences over the candidate and the number of elements in set of alternatives (A), $|A| \geq 3$, then every unanimous and non-manipulable voting rule is a dictatorship.

23.2 Stable matchings

The **stable matching problem** is the problem of finding a stable matching between two equally sized sets of elements given an preference order for each element. (We will be considering only equally sized sets). A matching is a bijection from the elements of one set to the elements of the other set. A matching is not stable if there exists a pair of elements, one from each set, which prefer each other to their assigned match.

That is, the matching $\mu_P : M \rightarrow W$ is **pairwise unstable** for a given preference profile P iff

$$\exists\, m,\ w,\ m',\ w' \text{ such that } \mu_P(m) = w,\ \mu_P(m') = w',\ w' P_m w,\ m P_{w'} m'$$

The notation $w' P_m w$ stands for, w' is more preferred than w in the preference order of m. We call (m, w') as a **blocking pair**.

23.2.1 Gale-Shapley, Deferred Acceptance (DA) algorithm

23.2.2 Example demonstrating DA algorithm

Let us understand the men-proposing version of the algorithm with this example.

Round 1:

1. Each unmatched man proposes to the most preferred woman who has not rejected him.
2. Each woman tentatively accepts the most preferred man from existing proposals.

This is repeated in the next rounds until all the men are matched.

Algorithm 1 Gale-Shapley Deferred Acceptance Algorithm

function MpStableMatching
  Input: List of men M and women W such that $|M| = |W| = n$, preference lists P_m and P_w for each man and woman
  Output: Stable matching S
  Initialize all $m \in M$ and $w \in W$ to be free
  while $\exists$ some m who is free **do**
    Choose such a man m
    Let w be the most preferred woman in m's preference list who has not rejected him yet
    if w is free **then**
      (m, w) is a tentative match
    else
      Let m' be the current partner of w
      if w prefers m over m' **then**
        (m, w) becomes a tentative match, m' becomes free
      else
        m remains free
      end if
    end if
  end while
  return S making all tentative matchings final
end function

$P_{m_1} : w_1 > w_4 > w_2 > w_3$ m_1

$P_{m_2} : w_3 > w_2 > w_4 > w_1$ m_2

$P_{m_3} : w_1 > w_2 > w_3 > w_4$ m_3

$P_{m_4} : w_2 > w_1 > w_4 > w_3$ m_4

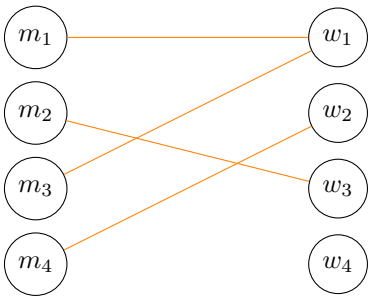
w_1 $P_{w_1} : m_2 > m_1 > m_4 > m_3$

w_2 $P_{w_2} : m_1 > m_2 > m_3 > m_4$

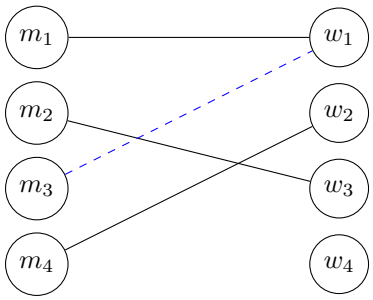
w_3 $P_{w_3} : m_3 > m_1 > m_2 > m_4$

w_4 $P_{w_4} : m_4 > m_2 > m_1 > m_3$

Figure 23.1: Example for DA algorithm

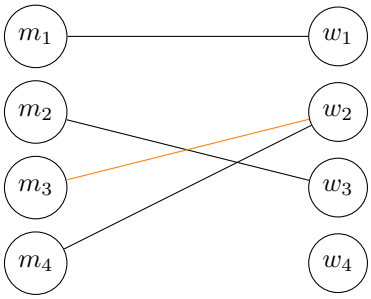


(a) Each unmatched man proposing to his most preferred woman [$m_1 - w_1$, $m_2 - w_3$, $m_3 - w_1$, $m_4 - w_2$]

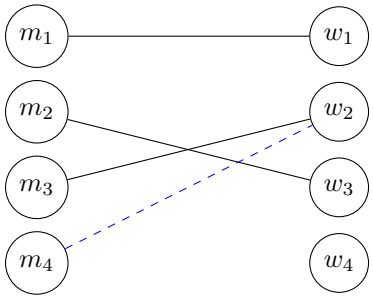


(b) Each woman tentatively accepting her most preferred man among the proposals [m_3 rejected by w_1]

Figure 23.2: Round 1

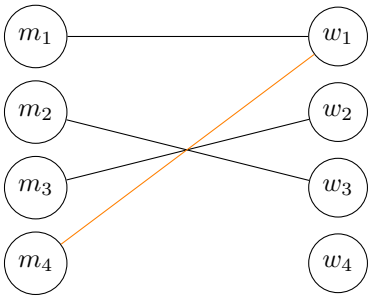


(a) Each unmatched man proposing to his most preferred woman $[m_3 - w_2]$

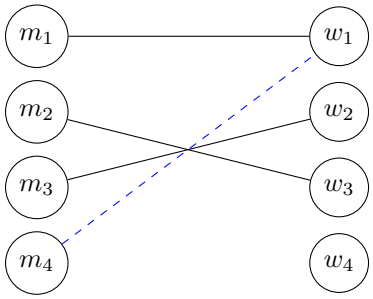


(b) Each woman tentatively accepting her most preferred man among the proposals [m_4 rejected by w_2]

Figure 23.3: Round 2

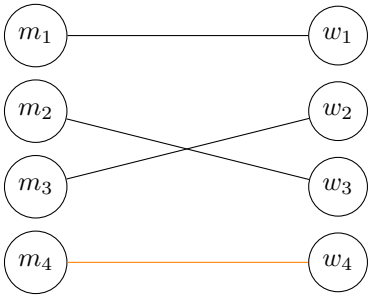


(a) Each unmatched man proposing to his most preferred woman $[m_4 - w_1]$

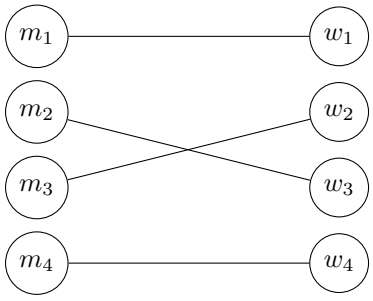


(b) Each woman tentatively accepting her most preferred man among the proposals [m_4 rejected by w_1]

Figure 23.4: Round 3



(a) Each unmatched man proposing to his most preferred woman $[m_4 - w_4]$



(b) Each woman tentatively accepting her most preferred man among the proposals [No rejections]

Figure 23.5: Round 4

23.2.3 Complexity, termination and correctness of the DA algorithm

Claim: DA algorithm terminates in polynomial time.
Proof:

- Every man makes atmost n proposals (since there are only n women and a man never proposes to a woman who has rejected him once).
- There are n men, and hence there can be atmost n^2 proposals before which the algorithm terminates. For each proposal, the algorithm does $\mathcal{O}(1)$ operations and comparisons (checking whether a man or a woman are free, comparing the preferences, and updating the tentative matches can all be done in constant number of operations, provided sufficient information is stored in appropriate data structures.)

  Hence, the complexity of the algorithm is $\mathcal{O}(n^2)$.

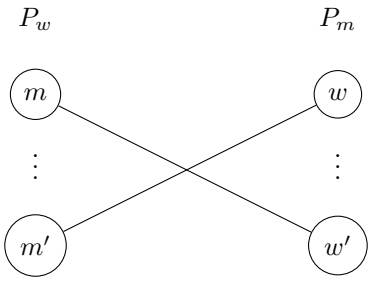
Claim: DA algorithm always returns a perfect matching (that is, all the vertices are matched).
Proof:

- No woman is matched to more than one man. So since there are an equal number of men and women, if a woman is unmatched, there will also be a man who is unmatched.
- Every woman is either not tentatively matched or gets multiple proposals and keeps one.
- Once a woman is tentatively matched, she is never unmatched again. (She will only shift her tentative match to one man to another man if she prefers the other man more than her current match.)
- The DA algorithm runs until all the men are matched. And when all men are matched, all the women will be matched to exactly one man, since no woman is matched to more than one man.

  Hence, the DA algorithm will return a perfect matching.

Claim: DA algorithm gives a pairwise stable matching.
Proof:

Suppose DA gives a pairwise unstable matching, that is, $\exists\ P$, blocking pair (m, w).



m is mapped to w' in the matching.
$\implies$ m had already been rejected by w. [$\because$ Each man proposes to women as per his preference order.]
$\implies$ w was tentatively matched to some man with higher preference than m in her preference order.

But, w is finally matched to m' which is below m in her preference order. This contradicts the working principle of the DA algorithm that change in the tentative matchings of women during the algorithm always increase their preference, whereas change in the tentative matchings of men during the algorithm always decrease their preference. Hence, our assumption was wrong and therefore, the algorithm always returns a pairwise stable matching.