Minimax: A Multiwinner Election Procedure

Rob LeGrand
legrand@cse.wustl.edu

Doctoral Research Seminar
12 November 2004
The Minimax Procedure: Problem

Outline

- Problem
- Two election procedures
  - Minisum
  - Minimax
- Computing minimax sets
  - Computational complexity
  - Heuristic for minimax
- Manipulating minimax
- Conclusions and future work
Problem

- \( A = \) set of \( k \) alternatives up for election
- \( B = \) set of \( n \) submitted approval ballots
- procedure should return a set \( W \subseteq A \) of winning alternatives
- number of winning alternatives can range from 0 to \( k \)
- one motivation (Brams et al.): multilateral treaties
Approval ballot example

010101

- voter approves three out of six alternatives \((b, d, f)\)
- voter’s most preferred outcome: 010101 \(\{b, d, f\}\)
- voter’s least preferred outcome: 101010 \(\{a, c, e\}\)
- voter prefers outcomes with smaller Hamming distances from 010101
- voter is indifferent among outcomes with equal Hamming distances from 010101, e.g. 000000 and 111111
Hamming distance

- used as measure of disagreement between a ballot and winner set
- Hamming distance between two sets $S$ and $T$:
  \[ d_H(S, T) = |S - T| + |T - S| \]
  
  - $d_H(\{a, b\}, \{a, c\}) = |\{b\}| + |\{c\}| = 2$
- Hamming distance between two bitstrings $S$ and $T$:
  \[ d_H(S, T) = |S \oplus T| \]
  
  - $d_H(010101, 111000) = |101101| = 4$
Outline

• Problem

• Two election procedures
  ○ Minisum
  ○ Minimax

• Computing minimax sets
  ○ Computational complexity
  ○ Heuristic for minimax

• Manipulating minimax

• Conclusions and future work
The minimun procedure

- simulates a majority yes/no vote on each alternative
  - alternative $a$ is elected iff more ballots approve than disapprove $a$

- equivalent to choosing the winner set $W$ with minimal "sumscore"
  - sumscore of a set is the total of the Hamming distances between the set and each ballot:

\[
\text{sumscore}(S) = \sum_{b \in B} d_H(S, b)
\]
Minisum example

\[ b_1 \quad 000011 \quad \{e, f\} \]
\[ b_2 \quad 000111 \quad \{d, e, f\} \]
\[ b_3 \quad 001011 \quad \{c, e, f\} \]
\[ b_4 \quad 010011 \quad \{b, e, f\} \]
\[ b_5 \quad 111100 \quad \{a, b, c, d\} \]
\[ W \quad 000011 \quad \{e, f\} \]

- \( e, f \) have 80% approval; \( b, c, d \) have 40% approval; \( a \) has 20% approval
- first four voters are quite satisfied with the minisum outcome
- last voter is completely dissatisfied and effectively ignored
The minimax procedure

- finds a winner set that minimizes the dissatisfaction of the least satisfied voters
- equivalent to choosing the winner set $W$ with minimal “maxscore”
  - maxscore of a set is the largest Hamming distance between the set and any ballot:

$$\text{maxscore}(S) = \max_{b \in B} d_H(S, b)$$
Minimax example

\[
\begin{align*}
   b_1 &\quad 000011 \quad \{e, f\} \\
   b_2 &\quad 000111 \quad \{d, e, f\} \\
   b_3 &\quad 001011 \quad \{c, e, f\} \\
   b_4 &\quad 010011 \quad \{b, e, f\} \\
   b_5 &\quad 111100 \quad \{a, b, c, d\} \\
   W &\quad 011111 \quad \{b, c, d, e, f\}
\end{align*}
\]

- all voters are relatively satisfied with the minimax outcome
- \( \text{maxscore}(W) = 3 \); all other sets have maxscore at least 4
Outline

- Problem
- Two election procedures
  - Minisum
  - Minimax
- Computing minimax sets
  - Computational complexity
  - Heuristic for minimax
- Manipulating minimax
- Conclusions and future work
Complexity

• finding a minimum set can be done in $O(kn)$ time
  ○ treat it as $k$ yes/no elections and report the majority winners

• finding a minimax set can be done in $O(2^k kn)$ time
  ○ brute-force approach calculates maxscore of each of $2^k$ possible winner sets and takes the lowest

• Can a minimax set be found in polynomial time?
A minimax variant: Fixed-size minimax

- like minimax, finds a winner set that minimizes the dissatisfaction of the least satisfied voters . . .
- but only considers sets of fixed size $m$
- likely more useful for real-world committee elections
- finding FSM winner set is provably NP-complete
Fixed-size minimax decision problem

**FIXED-SIZE MINIMAX (FSM)**

INSTANCE: Set $A$ with $|A| = k$; collection $B$ of $n$ subsets $B_1, B_2, \ldots B_n$ of $A$; nonnegative integer $d < k$; nonnegative integer $m \leq k$.

QUESTION: Is there a subset $W$ of $A$ such that $|W| = m$ and

$$d_H(W, B_i) = |W - B_i| + |B_i - W| \leq d$$

for all $i$?
**Vertex cover decision problem**

**VERTEX COVER (VC)**

INSTANCE: Graph $G = (V, E)$; positive integer $c \leq |V|$.

QUESTION: Is there a vertex cover of size $c$ or less for $G$, i.e., a subset $V' \subseteq V$ with $|V'| \leq c$ such that for each edge $E_i = u, v \in E$ at least one of $u$ and $v$ belongs to $V'$?
Vertex cover problem

Is there a vertex cover of size $\leq 3$?
The Minimax Procedure: Complexity

Vertex cover problem

{\(B, D, G\)} is a vertex cover of size 3.
### Equivalent FSM problem

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>Sequence</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>1100000</td>
<td>{a, b}</td>
</tr>
<tr>
<td>$b_2$</td>
<td>1001000</td>
<td>{a, d}</td>
</tr>
<tr>
<td>$b_3$</td>
<td>0110000</td>
<td>{b, c}</td>
</tr>
<tr>
<td>$b_4$</td>
<td>0001100</td>
<td>{d, e}</td>
</tr>
<tr>
<td>$b_5$</td>
<td>0011100</td>
<td>{c, d}</td>
</tr>
<tr>
<td>$b_6$</td>
<td>0001010</td>
<td>{d, f}</td>
</tr>
<tr>
<td>$b_7$</td>
<td>0010001</td>
<td>{c, g}</td>
</tr>
<tr>
<td>$b_8$</td>
<td>0000011</td>
<td>{f, g}</td>
</tr>
<tr>
<td>$W$</td>
<td>0101001</td>
<td>{b, d, g}</td>
</tr>
</tbody>
</table>
Reduction

- any vertex cover problem can be reduced to a fixed-size minimax problem
- the vertices of the VC graph become the FSM alternatives; the edges become the ballots
- each so-constructed ballot necessarily votes for exactly two alternatives since each edge is a set of exactly two vertices
- there is a vertex cover of size $\leq c \iff$ there is a FSM solution with maxscore $\leq d$
- minimax is NP-complete also (Frances & Litman)
Outline

- Problem
- Two election procedures
  - Minisum
  - Minimax
- Computing minimax sets
  - Computational complexity
  - Heuristic for minimax
- Manipulating minimax
- Conclusions and future work
The Minimax Procedure: Heuristic for minimax

Rob LeGrand

Heuristic for minimax

1. Choose ballots $b_1$ and $b_2$ such that $d_H(b_1, b_2)$ is maximized and $\text{sumscore}(b_1) \leq \text{sumscore}(b_2)$.

2. Start the current solution $s$ at $b_1$.

3. Repeat until $s$ does not change:
   (a) Initialize a collection $L$ of sets to be empty.
   (b) For each alternative $x$ on which $b_1$ and $b_2$ differ:
       - If $x \in s$, add $s - \{x\}$ to $L$.
       - If $x \notin s$, add $s \cup \{x\}$ to $L$.
   (c) Compare the sets in $L$ and, of the ones with the smallest maxscore, call the one with the smallest minisum score $t$. If $t$ has a smaller maxscore, or equal maxscore and smaller sumscore, than the current $s$, make it the new $s$.

4. Take $s$ as the minimax “solution”.
Minimax heuristic can fail

\[
\begin{align*}
  b_1 & : 000100 & \{d\} \\
  b_2 & : 101011 & \{a, c, e, f\} \\
  b_3 & : 011011 & \{b, c, e, f\} \\
  b_4 & : 100100 & \{a, d\} \\
  b_5 & : 010001 & \{b, f\} \\
  b_6 & : 100001 & \{a, f\} \\
  b_7 & : 111000 & \{a, b, c\} \\
  W & : 001000 & \{c\}
\end{align*}
\]

heuristic: 100100 \[\rightarrow\] 100000 \[\rightarrow\] 100001
Testing the heuristic

- heuristic only sometimes finds optimal minimax sets
- testing approach:
  - generate random ballots using uniform electorate model
  - find maxscores of optimal minimax set and set found by heuristic
  - scale heuristic maxscore so that optimal minimax set maxscore is 100% and $k$ (worst possible maxscore) is 0%
Heuristic performance

- higher percentage means closer to optimal on average

<table>
<thead>
<tr>
<th>$k \setminus n$</th>
<th>5</th>
<th>25</th>
<th>125</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>99.67%</td>
<td>92.71%</td>
<td>99.08%</td>
</tr>
<tr>
<td>12</td>
<td>99.23%</td>
<td>92.07%</td>
<td>94.70%</td>
</tr>
<tr>
<td>18</td>
<td>98.69%</td>
<td>93.14%</td>
<td>93.34%</td>
</tr>
</tbody>
</table>

- heuristic often finds optimal minimax set
- almost always finds set with near-optimal maxscore
The Minimax Procedure: Manipulating minimax

Outline

- Problem
- Two election procedures
  - Minisum
  - Minimax
- Computing minimax sets
  - Computational complexity
  - Heuristic for minimax
- Manipulating minimax
- Conclusions and future work
Manipulating minimax

sincere votes:

- $b_1 = 000011 \quad \{e, f\}$
- $b_2 = 000111 \quad \{d, e, f\}$
- $b_3 = 001011 \quad \{c, e, f\}$
- $b_4 = 010011 \quad \{b, e, f\}$
- $b_5 = 011111 \quad \{b, c, d, e, f\}$

- $W_1 = 000111 \quad \{d, e, f\}$
- $W_2 = 001011 \quad \{c, e, f\}$
- $W_3 = 010011 \quad \{b, e, f\}$

- all voters approve $e$ and $f$ and disapprove $a$
- voter 5 has Hamming distance 2 from each minimax winner set
Manipulating minimax

voter 5 is unscrupulous:

- $b_1 \ 000011 \ \{e, f\}$
- $b_2 \ 000111 \ \{d, e, f\}$
- $b_3 \ 001011 \ \{c, e, f\}$
- $b_4 \ 010011 \ \{b, e, f\}$
- $b_5 \ 111100 \ \{a, b, c, d\}$

$W \ 011111 \ \{b, c, d, e, f\}$

- by voting insincerely, voter 5 has manipulated the election to give his most preferred outcome decisively
Manipulating minimax

- if there are alternatives on which a voter is in the overwhelming majority, it may be able to vote safely against the majority on those alternatives to force more agreement with its relatively controversial choices
- it is reasonable to expect many minimax ballots to have been insincerely voted
- if all voters use the above strategy, minimax elections will become extremely unstable
Outline

- Problem
- Two election procedures
  - Minisum
  - Minimax
- Computing minimax sets
  - Computational complexity
  - Heuristic for minimax
- Manipulating minimax
- Conclusions and future work
Conclusions

- minimax minimizes maximum voter disagreement with the winner set
- can be seen as fairer than minisum
- finding an optimal minimax set is NP-complete
- good polynomial-time heuristics exist
- minimax is often easily manipulated
Future work

- Is it possible to find a better approach to a minimax heuristic? Can an approximation ratio be proved?
- How can the heuristics presented be modified to perform well on the fixed-size minimax problem?
- Is there a way to change minimax to lessen the effects of manipulability while retaining minimax’s coalition-forming properties?
References


thanks to Ron Cytron, Steven Brams and Morgan Deters

Questions?