Introducing a few firsts

• FIRST: Open Source project sponsored by CCSF/DT
• FIRST: Implementation of a Risk-Limiting Audit on a Ranked-Choice Voting contest
• FIRST: Independent validation of Dominion’s RCV Tabulation
What is a Risk-Limiting Audit?

- A Risk-Limiting Audit (RLA) offers a statistical guarantee:

  “If a full manual tally of the paper ballots would show that the reported election outcome is wrong, an RLA has a known minimum chance, the RLA limit, of leading to a full manual tally” – Philip B. Stark

  “As with other elections audit, the goal is to identify not only intentional alterations of ballots and tallies, but also bugs in election machines, such as software errors, scanners with blocked sensors or scanners skipping some ballots.” – Wikipedia
Two main types of RLA:

- **Ballot Polling**: Humans count a random sample of ballots and report any difference between manual percentage and computer percentage.

- **Ballot Comparison**: Election system provide a Cast Vote Record (CVR). Humans check physical ballots in a random sample of ballots. RLA system check results of human count against machine count.

Ballot comparison is more efficient than ballot polling due to its smaller sample size, and, arguably, less error-prone on an RCV.
Anyone is welcome to check the math

$$\text{audit-irv}(C, B, c_w, \alpha, \gamma)$$

1. \(\text{audits} \leftarrow \emptyset\)
2. \(F \leftarrow \emptyset \triangleright F\) is a set sequences to expand (the frontier)
3. \(LB \leftarrow 0\)
4. \(\triangleright \) Populate \(F\) with single-candidate sequences
5. \(\text{for each}(c \in C \setminus \{c_w\}):\)
6. \(\pi \leftarrow [c]\)
7. \(h \leftarrow \text{FindBestAudit}(\pi, C, B, c, \alpha, \gamma)\)
8. \(hy[\pi] \leftarrow h \triangleright \text{Record best hypothesis for } \pi\)
9. \(ba[\pi] \leftarrow \pi \triangleright \text{Record best ancestor sequence for } \pi\)
10. \(F \leftarrow F \cup \{\pi\}\)
11. \(\triangleright \) Repeatedly expand the sequence with largest ASN in \(F\)
12. \(\text{while}(|F| > 0):\)
13. \(\pi \leftarrow \text{argmax}\{\text{ASN}(hy[\pi]) \mid \pi \in F\}\)
14. \(F \leftarrow F \setminus \{\pi\}\)
15. \(\text{if(ASN}(hy[ba[\pi]]) \leq LB):\)
16. \(\text{audits} \leftarrow \text{audits} \cup \{hy[ba[\pi]]\}\)
17. \(F \leftarrow F \setminus \{\pi' \in F \mid ba[\pi] \text{ is a suffix of } \pi'\}\)
18. \(\text{continue}\)
19. \(\text{for each}(c \in C \setminus \pi):\)
20. \(\pi' \leftarrow [c] + \pi\)
21. \(h \leftarrow \text{FindBestAudit}(\pi', C, B, c, \alpha, \gamma)\)
22. \(hy[\pi'] \leftarrow h\)
23. \(ba[\pi'] \leftarrow \text{if } \text{ASN}(h) < \text{ASN}(hy[ba[\pi]]) \text{ then } \pi' \text{ else } ba[\pi]\)
24. \(\text{if } (|\pi'| = |C|):\)
25. \(\text{terminate} \text{ algorithm, full recount necessary}\)
26. \(\text{else:}\)
27. \(\text{audits} \leftarrow \text{audits} \cup \{hy[ba[\pi']])\}\)
28. \(LB \leftarrow \text{max}(LB, \text{ASN}(hy[ba[\pi']]))\)
29. \(F \leftarrow F \setminus \{\pi' \in F \mid ba[\pi] \text{ is a suffix of } \pi'\}\)
30. \(\text{continue}\)
31. \(\text{else:}\)
32. \(F \leftarrow F \cup \{\pi'\}\)
33. \text{return } \text{audits} \text{ with maximum ASN equal to } LB\]
Overview

• SHANGRLA is an Open Source RCV-RLA project sponsored by CCSF
  • https://github.com/pbstark/SHANGRLA
  • Implementation of academic research on RCV-RLA
  • Six key contributors across the globe on the project
    • We have yet to meet each other and we all spoke different languages:
      • English, American, Australian, Java, C++, Python, JSON, CSV, and Statistics

• SHANGRLA pilot is NOT an RLA
  • It is a ballot-comparison audit of Vote By Mail ballots (~68% of total)
  • A full RLA would require inclusion of all ballots casted
    • And a separate audit method for non VBM Ballots
Acknowledgements - CCSF -

CCSF would like to acknowledge the gargantuan efforts of the team:

CCSF Team:
- Linda J. Gerull CCSF, City CIO & Executive Director, Dept of Technologies
- Members of the San Francisco Dept of Elections

RCV Team:
- **Dr. Michelle Blom:**
  Research Fellow, School of Computing and Information Systems, The University of Melbourne, Australia
- **Dr. Andrew Conway:**
  CEO, Silicon Econometrics Pty. Ltd., Australia
- **Peter Stuckey:**
  Professor, Data Science & AI, Monash University, Melbourne, Australia
- **Vanessa Teague:**
  Associate Professor, School of Computing and Information Systems, The University of Melbourne, Australia

RLA Team:
- **Dan King:**
  CCSF Dept of Technology Consultant, CEO ViewPoint Technology, San Diego
- **Philip B. Stark:**
  Professor of Statistics, Associate Dean, Division of Mathematical and Physical Sciences, Regional Associate Dean (Interim), College of Chemistry and Division of Mathematical and Physical Sciences, University of Berkeley, CA
ShangRLA flow overview

1. Tabulates & Convert to RAIRE
   - CVR (JSON)
   - Raire (JSON)

2. Raire Assertion generator
   - Raire (JSON)

3. Assertions (Json)
   - Raire Assertion generator

4. Seed

5. Ballots to audit (CSV)
   - Manual Vote Recorder Tool
   - Manifest (tab)

6. Manual Vote Recorder Tool
   - CVR (json)
   - MVR (json)

7. Physical ballots

8. Audit results
   - Audit results
   - Elections Dept
   - Dominion
RLA Pilot - November 2019

Assertion Risk-Limiting Audit Workflow

Step 1 - Download Election Results
Download official election results
sfelections.sfgov.org
November 6, 2019 election results detailed report

Step 2 - Convert Election Results
Convert CVR into RAIRE format
github.com/raire/RAIRE-JSON

Step 3 - Generate RAIRE Assertions
Run the CVR through the RAIRE assertion generator
github.com/michaeleblom/audit-inv-cph/therearebranch

Step 4 - SHANGRLA Part I
Run Assertion Risk-Limiting Audit (assertion-RLA.ipynb)
steps 1-23
github.com/statesh/SHANGRLA

Step 5 - Manual Vote Record
Manually enter selected ballots
rla.ptek.io

Step 6 - SHANGRLA Part II
Run Assertion Risk-Limiting Audit (assertion-RLA.ipynb)
steps 27 onward
github.com/statesh/SHANGRLA

Step 7 - Review Results
Are the computed p values less than 0.05?

Yes - Audit Complete
Open RAIREExampleData.ipynb and view the elimination trees. Double-check that all branches with an alternative winner are pruned.

No - Repeat
Repeat with larger sample.
RAIRE Tabulator/Converter – Andrew Conway - Aus

Independent verification of Dominion’s tabulation

<table>
<thead>
<tr>
<th></th>
<th>Report 8</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loftus</td>
<td>59,762</td>
<td>66,228</td>
<td>83,511</td>
<td></td>
</tr>
<tr>
<td>Deutsch</td>
<td>26,901</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tung</td>
<td>37,161</td>
<td>46,403</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boudin</td>
<td>68,145</td>
<td>72,860</td>
<td>85,950</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City and County of San Francisco</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>SUZY LOFTUS</td>
</tr>
<tr>
<td>LEIF DAUTCH</td>
</tr>
<tr>
<td>NANCY TUNG</td>
</tr>
<tr>
<td>CHESA BOUDIN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer (Elimination)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assertions Generator – Michelle Blom - Aus

Many of the the 29 audits steps can be found here:

**Overview**

| Popularity | Repositories: 7 | Projects: 1 | Stars: 0 | Followers: 0 | Following: 0 |

**Popular repositories**

- **audit-irv-cp**
  - Code for generating and running ballot-level comparison audits for IRV elections.
  - C++  ★ 2

- **audit-irv-bp**
  - Code for generating and running ballot polling audits for IRV elections.
  - C++  ★ 1

- **margin-irv**
  - Margin computation for IRV elections.
  - C++

- **NSW2015**
  - Data files for use with a range of code available in my repositories. Can be used with: audit-irv-bp; audit-irv-cp.

- **STV-manipulator**
  - An implementation of two heuristics for computing candidate manipulations of Single Transferable Vote (STV) elections. These manipulations provide an upper bound on the margin of victory for such elections.
  - C++

- **AZUL**
  - Framework to support policy learning for the boardgame AZUL. The purpose of this framework is to allow students to implement algorithms for learning AI players for the game and evaluate the performance.
  - Python
Assertion Visualizer - Vanessa Teague – Aus.

Now print all the assertions. This gives you an explanation of the meaning of each one.

[3]:
```
printAssertions(wLoser,IRELims)
```

Not-Eliminated-Before assertions:
Confirmed: NEN 0: Candidate 15 cannot be eliminated before 16.

Not-Eliminated-Next assertions:
Confirmed: NEN 0: Candidate 15 cannot be eliminated next when ('16', '17') are eliminated.
Confirmed: NEN 1: Candidate 18 cannot be eliminated next when ('16', '15') are eliminated.
Confirmed: NEN 3: Candidate 15 cannot be eliminated next when ('16', '18') are eliminated.
RLA – Philip B. Stark – Berkeley

• Inventor of RLA
• Has participated in dozens of RLA
• His work is the basis for CORLA
• ShangRLA is a substantial update to CORLA’s methods

Was “busy” in November
Auditor Tool - Dan King - CCSF/San Diego

Reviewer #2 - Review and confirm (revise as needed)

Imprinted ID:
99808-81-1

1st Choice  2nd Choice  3rd Choice  4th Choice  5th Choice

- Suzy Loftus
- Leif Dautch
- Nancy Tung
- Chesa Boudin
- Write-in

No Consensus  Revise Selections  Confirmed
Next Step: Taking ShangRLA from pilot to product

• Phase I:
  • Standardize on languages
    • Transition out of Jupyter notebook
  • Migrate from files to an RDBMS
    • JSON is ill-suited for a system that has a natural entity-relationship model
  • Build a test suite above and beyond unit tests
  • Document

• Phase II:
  • Support for Multi-Contest auditing
  • Integrate non-VBM Ballot auditing
  • Enhance the UI
  • ShangRLA is engineered to support various forms of contest beyond RCV
    • ...but “official support” may require further development and testing