20 Questions

Twenty (Simple) Questions

Yuval Dagan, Yuval Filmus, Ariel Gabizon, Shay Moran
**Twenty Questions Game**

**Alice**

- Thinks of an object according to known distribution $\mu$

**Bob**

- Finds object using Yes/No questions
- Attempts to minimize expected # of questions

**Diagram:**
- **Animate?**
  - Yes
  - No
- **Human?**
  - Yes: Alice
  - No: Cat
  - Yes: Mat
Twenty Questions Game

Alice
- Thinks of an object according to known distribution \( u \)

Bob
- Finds object using Yes/No questions
- Attempts to minimize expected # of questions

Optimal algorithm: Huffman coding (1952)

While more than one object remains:
- Repeatedly merge two least probable objects

Cost: between \( H(\mu) \) and \( H(\mu) + 1 \)

Issue: Huffman’s algorithm can ask arbitrary questions

Challenge: Same performance using fewer questions
## Results at a glance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Questions</th>
<th>Number</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huffman ‘52</td>
<td>Arbitrary</td>
<td>$2^n$</td>
<td>entropy + 1</td>
</tr>
<tr>
<td>Gilbert–Moore ‘59</td>
<td>$&lt;$</td>
<td>$n$</td>
<td>entropy + 2</td>
</tr>
<tr>
<td>Rissanen ‘73</td>
<td>$&lt;$</td>
<td>$n$</td>
<td>entropy + 2</td>
</tr>
<tr>
<td>this paper</td>
<td>$\leq$</td>
<td>$2n$</td>
<td>entropy + 1</td>
</tr>
<tr>
<td>this paper</td>
<td>base $n^{1/r} \leq$</td>
<td>$rn^{1/r}$</td>
<td>entropy + $r$</td>
</tr>
<tr>
<td>this paper</td>
<td>non-constructive</td>
<td>$1.25^n$</td>
<td>Huffman</td>
</tr>
<tr>
<td>this paper</td>
<td>$\subseteq[n/2], \supseteq[n/2]$</td>
<td>$1.41^n$</td>
<td>Huffman</td>
</tr>
<tr>
<td>this paper</td>
<td>intervals with holes</td>
<td>$n^{O(1/\varepsilon)}$</td>
<td>Huffman+$\varepsilon$</td>
</tr>
</tbody>
</table>

Most of our results – optimal with respect to number of questions!
Gilbert–Moore vs Rissanen

\[ P(x_1) = \frac{1}{5}, \; P(x_2) = \frac{1}{5}, \; P(x_3) = \frac{1}{4}, \; P(x_4) = \frac{3}{20}, \; P(x_5) = \frac{1}{5} \]

Binary search on \([0,1]\)
Equivalent to arithmetic coding
Obtaining redundancy 1

Problem: $x_1 \quad x_2 \quad x_3$

Requires two “<” questions to isolate!

Solution: also allow “=” queries!

Rissanen

While there is more than one live element:
Ask the most balanced “<” question

Our algorithm

While there is more than one live element:
Let $x_{\text{max}}$ be most probable live element
If $P(x_{\text{max}}) \geq 0.3$: Ask “$x = x_{\text{max}}$?”
Otherwise: Ask most balanced “<” question
Outline of analysis

- Let $R(\mu) = Alg(\mu) - H(\mu) - 1$.
  Our goal: show that $R(\mu) \leq 0$ for all $\mu$.

- Write a recurrence relation for $R(\mu)$ in terms of $\mu|_{\text{Yes}}$ and $\mu|_{\text{No}}$.
  Use $R(\mu|_{\text{Yes}}), R(\mu|_{\text{No}}) \leq 0$ to obtain an upper bound on $R(\mu)$.

- Let $r(\rho) = \max$ of $R(\mu)$ in terms of prob of most likely element.
  Our goal: show that $r(\rho) \leq 0$ for all $\rho$.

- Write a recurrence relation upper-bounding $r(\rho)$.

- Solve the recurrence relation to finish the proof.
Questions – redundancy tradeoff

Our algorithm uses $2n$ potential question to guarantee redundancy 1. How many questions are needed to guarantee redundancy $r$?

Idea: Write index $i$ of unknown element in base $n^{1/r}$: $i = i_{r-1} \ldots i_0$.

Use redundancy 1 algorithm to determine $i_{r-1}, \ldots, i_0$ one by one.

The algorithm uses $2rn^{1/r}$ potential questions to guarantee redundancy $r$.

Matching lower bound $\Omega(rn^{1/r})$:

Consider distributions concentrated on single element (entropy $\approx 0$).

Must be able to isolate each element using $r$ questions.
Some open questions

• How fast can we find optimal search tree using “<” and “=”?
  The best search tree using “<” (i.e., BST) can be found in $O(n \log n)$.
  In contrast, the best known algorithm when allowing both “<” and “=” takes $O(n^4)$.

• How many questions are needed to guarantee redundancy 1?
  Our results: between $n$ and $2n$.

• What happens if answerer can lie $k$ times?
  Work in progress: can achieve redundancy $k \sum \mu_i \log \log (1/\mu_i) + \tilde{O}(k^2)$.

• What if we assume that all probabilities are small?
  Classical result of Gallager: can’t go below 0.086 in worst case even for Huffman code.
  Preliminary results: answer for “<” and “=” queries is between 0.501 and 0.586.

• Generalize the theory to $d$-way queries.