A Coding Theoretic Framework for Query Learning

Clayton Scott

Electrical Engineering and Computer Science
University of Michigan

Collaborators: Suresh Bhavnani and Gowtham Bellala
Toxic Chemical Emergency

Hundreds of toxic chemical incidents per year (Kleindorfer et al., 2003)
Chemical Identification

Needed to treat victims, decontaminate site, issue neighborhood warnings, etc.
Decision Support for Chemical Identification
WISER Database

*Wireless Information System for Emergency Responders*

- Maintained by NLM, panel of chemists/toxicologists
- Lists which symptoms are caused by which chemicals
- Represented as bipartite network, or binary table

![Bipartite Network Illustration]

<table>
<thead>
<tr>
<th>chemicals</th>
<th>symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1 0 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1 1 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 1 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>
Network Layout of WISER Database

- ~ 300 chemicals, 80 symptoms
- Edge density ~ 0.4, symptoms tend to be non-specific

Bhavnani, et al., 2007
WISER

Wireless Information System for Emergency Responders

User selects a symptom

Non-matching chemicals eliminated

Symptom nonspecificity Unguided searching → Too many symptoms needed
Query Learning

**Objects:** $\Theta = \{\theta_1, \ldots, \theta_M\}$

**Queries:** $Q = \{q_1, \ldots, q_N\}$

<table>
<thead>
<tr>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$\pi_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_6$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Problem statement

- object selected at random
- determine object with as few queries as possible

$$\sum_{i=1}^{M} \pi_i = 1$$
Other Applications of Query Learning

Objects
- chemicals
- network failures
- faults
- classifiers
- ...

Queries
- symptoms
- network measurements
- alarms
- labels at specific points (active learning)
- ...

(active learning)
Outline

• Connecting query learning to source coding
• Extensions
  ❑ Exponentially weighted query costs
  ❑ Group identification
  ❑ Query noise
• Not in this talk
  ❑ Multiple objects present
  ❑ Likelihoods, Bayesian networks
  ❑ Human factors, usability, etc.
Decision Trees

Generalized binary search:
- Greedy, top-down algorithm
- Select query that balances remaining objects

\[
E[\# \text{ of queries}] = \pi_1 \cdot 2 + \pi_2 \cdot 4 + \pi_3 \cdot 2 + \pi_4 \cdot 4 + \pi_5 \cdot 3 + \pi_6 \cdot 2
\]
Source Coding

Fixed alphabet: \( \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\} \)
Prior probabilities: \( \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6 \)
**Goal:** efficient binary encoding

\[ \theta_3 \theta_2 \theta_5 \theta_3 \theta_6 \theta_2 \theta_1 \theta_3 \theta_1 \theta_3 \theta_3 \ldots \xrightarrow{\text{encoder}} 1110101001101101101000 \ldots \]

Instant decoding \( \implies \text{prefix code} \)

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
<th>codelength ( l_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_1 )</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>( \theta_2 )</td>
<td>1010</td>
<td>4</td>
</tr>
<tr>
<td>( \theta_3 )</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>( \theta_4 )</td>
<td>1011</td>
<td>4</td>
</tr>
<tr>
<td>( \theta_5 )</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>( \theta_6 )</td>
<td>01</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ E[\text{codelength}] = \sum_{i} \pi_i l_i \]
Source Coding

- $E[\text{codelength}] \geq - \sum_{i} \pi_i \log_2 \pi_i$
  
  $H_1(\{\pi_i\})$, Shannon entropy

- Huffman coding (Huffman, 1952)
  - Optimal
  - Bottom-up
  - Doesn’t generalize to query learning

- Shannon-Fano coding (Shannon, 1948; Fano, 1961)
  - Suboptimal
  - Top-down
  - Generalizes to query learning $\rightarrow$ GBS
Source Coding vs. Query Learning

- Same goal: minimize expected codelength / # of queries

  \[ \implies E[\# \text{ of queries}] \geq H_1(\{\pi_i\}) \]

- Query learning does not allow arbitrary trees

<table>
<thead>
<tr>
<th>( \theta )</th>
<th>( q_1 )</th>
<th>( q_2 )</th>
<th>( q_3 )</th>
<th>( q_4 )</th>
<th>( q_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_1 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \theta_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \theta_3 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \theta_4 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \theta_5 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \theta_6 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

  \[ \implies \text{only 5 possible splits} \]  
  (versus 31 for source coding)

- In query learning, finding optimal tree is NP-complete

  \[ \implies \text{need suboptimal algorithms} \]
Exact formula for arbitrary tree/code

Theorem: For any decision tree $T$, 

$$E[\# \text{ of queries}] = H_1(\{\pi_i\}) + \sum_{a \in \text{interior}(T)} \pi_a[1 - H(\rho_a)]$$

where 

$$\pi_a := \sum_{i : i \text{ reaches node } a} \pi_i$$
$$H(\rho) := -\rho \log_2 \rho - (1 - \rho) \log_2 (1 - \rho)$$
$$\rho_a := \frac{\pi_{\text{leftchild}(a)}}{\pi_a}$$

$$\rho_a = \frac{.1}{.1 + .05 + .1} = .4$$
Query Learning as Greedy Optimization

\[ E[\# \text{ of queries}] = H_1(\{\pi_i\}) + \sum_{a \in \text{interior}(T)} \pi_a[1 - H(\rho_a)] \]

Top-down, greedy optimization

\[ \implies \text{maximize } H(\rho_a) \]

\[ \implies \text{minimize } |\rho_a - \frac{1}{2}| \]

\[ \implies \text{generalized binary search} \]

<table>
<thead>
<tr>
<th></th>
<th>(q_1)</th>
<th>(q_2)</th>
<th>(q_3)</th>
<th>(q_4)</th>
<th>(q_5)</th>
<th>(\pi_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta_1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.25</td>
</tr>
<tr>
<td>(\theta_2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.05</td>
</tr>
<tr>
<td>(\theta_3)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.3</td>
</tr>
<tr>
<td>(\theta_4)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
</tr>
<tr>
<td>(\theta_5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
</tr>
<tr>
<td>(\theta_6)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.2</td>
</tr>
</tbody>
</table>
Outline

• Extensions of generalized binary search
  - Exponentially weighted query costs
  - Group identification
  - Query noise
Exponentially Weighted Costs

For $\lambda \geq 1$, minimize

$$\log_\lambda \left( \sum_{i=1}^{M} \pi_i \lambda^{d_i} \right)$$

where $d_i =$ depth of $\theta_i$

- $\lambda \to 1 \implies$ average depth
- $\lambda \to \infty \implies$ maximum depth (worst case)
- Source coding (arbitrary trees allowed) $\implies$ efficient optimal algorithm
- Query learning $\implies$ no efficient optimal algorithm
Rényi Entropy

**Lower bound** (Campbell, 1956): For any $\lambda$ and any tree

$$\log_\lambda \left( \sum_{i=1}^{M} \pi_i \lambda^{d_i} \right) \geq H_\alpha(\{\pi_i\})$$

where

$$H_\alpha(\{\pi_i\}) = \frac{1}{1 - \alpha} \log_2 \left( \sum_{i=1}^{M} \pi_i^\alpha \right)$$

and $\alpha = \frac{1}{1 + \log_2 \lambda}$
Exact Formula for Exponential Costs

**Theorem:** For any fixed $\lambda \geq 1$, and any tree $T$,

$$
\sum_{i=1}^{M} \pi_i \lambda^{d_i} = \lambda^{H_\alpha(\{\pi_i\})} + \sum_{a \in \text{int}(T)} \pi_a \left[ (\lambda - 1)\lambda^{d_a} - D_\alpha + \rho_a D_{\text{left}(u)}^\alpha + (1 - \rho_a) D_{\text{right}(u)}^\alpha \right]
$$

where

$$
D_\alpha^\alpha := \left[ \sum_{i : i \text{ reaches node } a} \left( \frac{\pi_i}{\pi_a} \right)^{\alpha} \right]^{1/\alpha}
$$

Greedy, top-down algorithm: $\lambda$-GBS
Results: WISER Database

300 chemicals, 80 symptoms, $\pi_i \propto i^{-1}$
Group Identification

Example: Identify class to which chemical belongs (pesticide, poison, etc.)

<table>
<thead>
<tr>
<th>$\theta_i$</th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$\pi_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.25</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.05</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.3</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
</tr>
<tr>
<td>$\theta_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
</tr>
<tr>
<td>$\theta_6$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.2</td>
</tr>
</tbody>
</table>

Best tree

$E[\# \text{ of queries}] = 1$

GBS

$E[\# \text{ of queries}] = 2.4$
## Group Identification

<table>
<thead>
<tr>
<th></th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$\pi_i$</th>
<th>$y_i$</th>
<th>$\tilde{\pi}_1$</th>
<th>$\tilde{\pi}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.25</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.05</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_6$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

labels $y_i \in \{1, 2, \ldots, K\}$

**Theorem:** For any tree $T$,

$$E[\text{# of queries}] = H_1(\{\tilde{\pi}_k\}) + \sum_{a \in \text{interior}(T)} \pi_a \left[ 1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_a^k}{\pi_a} H(\rho_a^k) \right]$$
Group-GBS

**Greedy algorithm:** At each successive node, choose query to minimize

\[ 1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_a^k}{\pi_a} H(\rho_a^k) \]

This prefers queries such that

- \( \rho_a \approx \frac{1}{2} \implies \text{balanced trees} \)
- \( \rho_a^k \approx 0 \text{ or } 1 \) for each \( k \implies \text{preserve groups} \)
Group Identification Results

- WISER database (300 chemicals, 80 symptoms)
- 16 chemical classes (pesticide, poison, corrosive acid, etc.)
- uniform prior on chemicals

<table>
<thead>
<tr>
<th>Entropy lower bound</th>
<th>3.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group-GBS</td>
<td>7.79</td>
</tr>
<tr>
<td>GBS</td>
<td>7.95</td>
</tr>
<tr>
<td>Random Search</td>
<td>16.33</td>
</tr>
</tbody>
</table>

- WISER-like database with better “concordance” within classes

<table>
<thead>
<tr>
<th>Entropy lower bound</th>
<th>3.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group-GBS</td>
<td>3.50</td>
</tr>
<tr>
<td>GBS</td>
<td>7.51</td>
</tr>
<tr>
<td>Random Search</td>
<td>16.12</td>
</tr>
</tbody>
</table>
Query Noise

- Suppose $\theta_2$ is the true object

<table>
<thead>
<tr>
<th></th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$q_6$</th>
<th>$q_7$</th>
<th>$q_8$</th>
<th>$q_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Ideal query responses:

| $\theta_2$ | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

- Noisy query responses:

| $\theta_2$ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
Query Noise

- **Nearest neighbor** decoding: If

\[
\min_{i \neq j} \ d_{\text{Hamming}}(\theta_i, \theta_j) \geq \epsilon
\]

can recover

\[
\delta = \left\lfloor \frac{\epsilon - 1}{2} \right\rfloor
\]

query errors

\[
\begin{array}{cccccccccc}
\theta_1 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
\theta_2 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
\theta_3 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
\theta_4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\end{array}
\]

\[
\epsilon = 5 \implies \delta = 2
\]
Query Noise

<table>
<thead>
<tr>
<th>$\theta_1$</th>
<th>1 0 1 1 0 1 1 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_2$</td>
<td>0 1 1 0 1 0 0 1 0</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>0 1 0 1 0 0 1 1 1</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>1 1 0 0 1 1 1 0 1</td>
</tr>
</tbody>
</table>

Now apply Group-GBS

\[
\{ \theta : d_{\text{Hamming}}(\theta, \theta_1) \leq \delta \}
\]

\[
\{ \theta : d_{\text{Hamming}}(\theta, \theta_2) \leq \delta \}
\]

\[
\{ \theta : d_{\text{Hamming}}(\theta, \theta_3) \leq \delta \}
\]

\[
\{ \theta : d_{\text{Hamming}}(\theta, \theta_4) \leq \delta \}
\]
Query Noise Results

- Modified WISER database ($\epsilon = 5, \delta = 2$)
Conclusion

Summary
• Query learning = constrained source coding
• Exact formulas for performance ➞ greedy algorithms
• Exponential costs, group identification, query noise

Work in progress
• Likelihoods (fully Bayesian model)
• Ranking
• User confidence
• Multiple objects present