A Coding Theoretic Framework for Query Learning

Clayton Scott

Electrical Engineering and Computer Science
University of Michigan
Collaborators

Gowtham Bellala

Suresh Bhavnani

Max Yi Ren

Panos Papalambros
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 Diagram]

<table>
<thead>
<tr>
<th>chemicals</th>
<th>symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>chemicals</td>
<td>symptoms</td>
</tr>
<tr>
<td>1 0 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>1 1 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 nonspecific

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

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

<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>$\pi_1$</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$\pi_2$</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\pi_3$</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$\pi_4$</td>
</tr>
<tr>
<td>$\theta_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$\pi_5$</td>
</tr>
<tr>
<td>$\theta_6$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\pi_6$</td>
</tr>
</tbody>
</table>

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

**Objects**
- chemicals
- network failures
- faults
- classifiers
- …

**Queries**
- symptoms
- network measurements
- alarms
- labels at specific points (active learning)
- …
Outline

• Connecting query learning to source coding
• Generalizations
  ❑ Exponentially weighted costs
  ❑ Group identification
• Application to
  ❑ Query noise
  ❑ Preference elicitation
• Not in this talk
  ❑ Multiple objects present
  ❑ Likelihoods, Bayesian networks
  ❑ Network fault detection
  ❑ 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

\[
\theta_3\theta_2\theta_5\theta_3\theta_6\theta_2\theta_1\theta_3\theta_1\theta_3\theta_3 \ldots \rightarrow \text{encoder} \rightarrow 11101001101101101000\ldots
\]

Instant decoding \implies \textbf{prefix} code

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
<th>codelength (\ell_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>

\[
E[\text{codelength}] = \sum_i \pi_i \ell_i
\]
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
  \[ E[\# \text{ of queries}] \geq H_1(\{\pi_i\}) \]

- Query learning does not allow arbitrary trees

<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>
</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>

\[ \Rightarrow \text{only 5 possible splits} \]

(versus 31 for source coding)

- In query learning, finding optimal tree is NP-complete
  \[ \Rightarrow \text{need suboptimal algorithms} \]
**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>(\theta)</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>
Exponentially Weighted Costs

For \( \lambda \geq 1 \), minimize

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

where \( d_i = \text{depth of } \theta_i \)

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

**Lower bound** (Campbell, 1956): For any $\lambda > 1$ 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 $T$,

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

where

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

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

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

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

GBS

Best tree

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

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

<table>
<thead>
<tr>
<th>$\theta_1$</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>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.25</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.05</td>
<td>2</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.3</td>
<td>1</td>
</tr>
<tr>
<td>$\theta_5$</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>$\theta_6$</td>
<td>0</td>
<td>0</td>
<td>0</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>
</tr>
</tbody>
</table>

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

$\tilde{\pi}_1 = .75$

$\tilde{\pi}_2 = .25$

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

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

- \( \rho_a \approx \frac{1}{2} \implies \text{balanced trees} \)
- \( \rho_a^k \approx 0 \text{ or } 1 \text{ for each } k \implies \text{preserve groups} \)
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>
Performance Guarantee

Alternate greedy algorithm: At each successive node, choose query to maximize

$$\pi_l(a)\pi_r(a) - \sum_{k=1}^{K} \frac{\pi_a^k}{\pi_a^l(a)\pi_r(a)}$$

Similar intuition as previous criterion.

Theorem: Let $\hat{T}$ denote the tree based on the above splitting criterion, and let $T^*$ be the tree with minimum expected depth. Then

$$\mathbb{E}[\text{depth}(\hat{T})] \leq \left(2\ln\left(\frac{1}{\sqrt{3}\pi_{\text{min}}}ight) + 1\right)\mathbb{E}[\text{depth}(T^*)]$$

where $\pi_{\text{min}} = \min_i \pi_i$. 
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

<table>
<thead>
<tr>
<th></th>
<th>q₁</th>
<th>q₂</th>
<th>q₃</th>
<th>q₄</th>
<th>q₅</th>
<th>q₆</th>
<th>q₇</th>
<th>q₈</th>
<th>q₉</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ₁</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>θ₂</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>θ₃</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>θ₄</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\(\epsilon = 5 \implies \delta = 2\)
Query Noise

Now apply Group-GBS

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
</tr>
</tbody>
</table>

\[
\{ \theta : d_{\text{Hamming}}(\theta, \theta_1) \leq \delta \} \\
\{ \theta : d_{\text{Hamming}}(\theta, \theta_2) \leq \delta \} \\
\{ \theta : d_{\text{Hamming}}(\theta, \theta_3) \leq \delta \} \\
\{ \theta : d_{\text{Hamming}}(\theta, \theta_4) \leq \delta \}
\]
Query Noise Results

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

• Given several designs for a product
  
  ❑ E.g., laptop computers: each design has different combinations of features: memory, weight, cost, size, battery life, etc.

• Choose the most preferred design for a population of users

• Survey the population; Each survey consists of a sequence of pairwise comparisons
  
  ❑ “Do you prefer Design A or Design B?”

• Goal: Construct survey to determine most preferred design using the minimal number of queries
Preference Elicitation as Group Identification

- Products $\mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{R}^d$

- Each user has a ranking of products, e.g.,

  $$\theta = \mathbf{x}_3 \prec \mathbf{x}_7 \prec \mathbf{x}_2 \prec \cdots$$

- Set of possible rankings

  $$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\}$$

- Group rankings by most preferred design

  $$\Theta^k_i := \{\theta \in \Theta \mid x_k \text{ most preferred}\}$$
Pairwise Comparisons are Queries

<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>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</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>
<tr>
<td>$\vdots$</td>
<td>\vdots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This ranking is consistent with this query

This ranking is not consistent with this query
Some Interesting Features

- Initial probability distribution $\pi_i$ on rankings: uniform
- Users surveyed sequentially, so can update distribution on rankings after each survey
- Estimating probability $\pi_a$, where $a$ is a node in the tree, is nontrivial
- Assume *partworth model*: Each user is represented by a vector $\mathbf{w} \in \mathbb{R}^D$, and pairwise comparison between $\mathbf{x}_i$ and $\mathbf{x}_j$ depends on the sign of
  \[
  \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j).
  \]
  This enables geometric approximations of $\pi_a$ quantities (details omitted but it’s an SVM type algorithm)
Conclusion

- Query learning = constrained source coding
- Exact formulas for performance → greedy algorithms
- Group identification → preference elicitation