Feature selection

- Given random variables $Y, X_1, \ldots, X_n$
- Want to predict $Y$ from subset $X_A = (X_{i_1}, \ldots, X_{i_k})$

Want $k$ most informative features:

$$A^* = \text{argmax } IG(X_A; Y) \text{ s.t. } |A| \leq k$$

where $IG(X_A; Y) = H(Y) - H(Y | X_A)$

Uncertainty before knowing $X_A$  Uncertainty after knowing $X_A$

Problem inherently combinatorial!
Factoring distributions

- Given random variables $X_1, ..., X_n$
- Partition variables $V$ into sets $A$ and $V \setminus A$ as independent as possible

Formally: Want

$$A^* = \arg\min_A I(X_A; X_{V \setminus A}) \quad \text{s.t.} \quad 0 < |A| < n$$

where $I(X_A, X_B) = H(X_B) - H(X_B \mid X_A)$

Fundamental building block in structure learning
[Narasimhan&Bilmes, UAI ’04]

Problem inherently combinatorial!
Combinatorial problems in ML

Given a (finite) set $V$, function $F: 2^V \rightarrow \mathbb{R}$, want

$$A^* = \text{argmin } F(A) \quad \text{s.t. some constraints on } A$$

Solving combinatorial problems:

- Mixed integer programming?
  
  Often difficult to scale to large problems

- Relaxations? (e.g., L1 regularization, etc.)
  
  Not clear when they work

- This talk:
  
  Fully combinatorial algorithms (spanning tree, matching, ...)
  
  Exploit problem structure to get guarantees about solution!
Set functions

- Finite set $V = \{1, 2, \ldots, n\}$
- Function $F: 2^V \rightarrow \mathbb{R}$
- Will always assume $F(\emptyset) = 0$ (w.l.o.g.)
- Assume black-box that can evaluate $F$ for any input $A$

Example: $F(A) = IG(X_A; Y) = H(Y) - H(Y | X_A)$

$$= \sum_{y,x_A} P(x_A) \left[ \log P(y | x_A) - \log P(y) \right]$$

- $F(\{1, 2\}) = 0.9$
- $F(\{2, 3\}) = 0.5$
Submodular set functions

- Set function $F$ on $V$ is called **submodular** if
  
  For all $A, B \subseteq V$: $F(A) + F(B) \geq F(A \cup B) + F(A \cap B)$

- Equivalent **diminishing returns** characterization:

  **Submodularity:**
  
  For $A \subseteq B, s \notin B$, $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$
Submodularity and supermodularity

Set function \( F \) on \( V \) is called **submodular** if

1) For all \( A,B \subseteq V \): \( F(A)+F(B) \geq F(A \cup B)+F(A \cap B) \)

\( \Leftrightarrow \) 2) For all \( A \subseteq B \), \( s \not\in B \), \( F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B) \)

\( F \) is called **supermodular** if \( -F \) is submodular

\( F \) is called **modular** if \( F \) is both sub- and supermodular

for modular ("additive") \( F \), \( F(A) = \sum_{i \in A} w(i) \)
Example: Set cover

Place sensors in building

Node predicts values of positions with some radius

Want to cover floorplan with discs

For \( A \subseteq V \): \( F(A) = \) “area covered by sensors placed at \( A \)”

Formally:

\[
W \text{ finite set, collection of } n \text{ subsets } S_i \subseteq W
\]

For \( A \subseteq V = \{1, ..., n\} \) define \( F(A) = | \bigcup_{i \in A} S_i | \)
Set cover is submodular

\[ A = \{S_1, S_2\} \]

\[ B = \{S_1, S_2, S_3, S_4\} \]

\[ F(A \cup \{S'\}) - F(A) \geq F(B \cup \{S'\}) - F(B) \]
Example: Mutual information

- Given random variables $X_1, \ldots, X_n$
  
  $F(A) = I(X_A; X_{V \setminus A}) = H(X_{V \setminus A}) - H(X_{V \setminus A} | X_A)$

Lemma: Mutual information $F(A)$ is submodular

$F(A \cup \{s\}) - F(A) = H(X_s | X_A) - H(X_s | X_{V \setminus (A \cup \{s\})}$

Nonincreasing in $A$: $A \subseteq B \Rightarrow H(X_s | X_A) \geq H(X_s | X_B)$

$\delta_s(A) = F(A \cup \{s\}) - F(A)$ monotonically nonincreasing

$\iff F$ submodular 😊
Example: Influence in social networks
[Kempe, Kleinberg, Tardos KDD ’03]

Who should get free cell phones?

\[ V = \{\text{Alice}, \text{Bob}, \text{Charlie}, \text{Dorothy}, \text{Eric}, \text{Fiona}\} \]

\[ F(A) = \text{Expected number of people influenced when targeting } A \]
Influence in social networks is submodular [Kempe, Kleinberg, Tardos KDD ’03]

Key idea: Flip coins $c$ in advance $\rightarrow$ “live” edges

$$F_c(A) = \text{People influenced under outcome } c \text{ (set cover!)}$$

$$F(A) = \sum_c P(c) F_c(A) \text{ is submodular as well!}$$
Closedness properties

$F_1, \ldots, F_m$ submodular functions on $V$ and $\lambda_1, \ldots, \lambda_m > 0$

Then: $F(A) = \sum_i \lambda_i F_i(A)$ is submodular!

Submodularity closed under nonnegative linear combinations!

Extremely useful fact!!

- $F_\theta(A)$ submodular $\Rightarrow \sum_\theta P(\theta) F_\theta(A)$ submodular!
- Multicriterion optimization:
  $F_1, \ldots, F_m$ submodular, $\lambda_i \geq 0$ $\Rightarrow \sum_i \lambda_i F_i(A)$ submodular
Example: Greedy algorithm for feature selection

Given: finite set $V$ of features, utility function $F(A) = IG(X_A; Y)$

Want: $A^* \subseteq V$ such that

$A^* = \underset{|A| \leq k}{\text{argmax}} F(A)$

NP-hard!

Greedy algorithm:

Start with $A = \emptyset$

For $i = 1$ to $k$

$s^* := \underset{s}{\text{argmax}} F(A \cup \{s\})$

$A := A \cup \{s^*\}$

How well can this simple heuristic do?
Why is submodularity useful?

**Theorem** [Nemhauser et al.’78]

Greedy maximization algorithm returns $A_{\text{greedy}}$:

$$F(A_{\text{greedy}}) \geq (1-1/e) \max_{|A| \leq k} F(A)$$

Greedy algorithm gives near-optimal solution!

More details and exact statement later

For info-gain: Guarantees best possible unless P = NP!

[Krause, Guestrin UAI ’05]
Submodularity in Machine Learning

Several problems in Machine Learning are submodular:

- **Minimization:** $A^* = \arg\min F(A)$
  - Structure learning ($A^* = \arg\min I(X_A; X_{V\setminus A})$)
  - Clustering
  - MAP inference in Markov Random Fields
  - ...

- **Maximization:** $A^* = \arg\max F(A)$
  - Feature selection
  - Active learning
  - Ranking
  - ...

$\Box$
Submodular Function Minimization

Ellipsoid Method

$O(n^5)$

Grötschel, Lovász, Schrijver (1981, 1988)

Cunningham (1985)

$O(n^7)$

Iwata, Fleischer, Fujishige (2000)

Schrijver (2000)

$n^8$

$O(n^5)$

Iwata, Orlin (2009)

Fully Combinatorial

$n$ = number of elements in $V$

Iwata (2002)

Fleischer, Iwata (2000)

Orlin (2007)
Symmetric Submodular Functions

\[ f : 2^V \rightarrow \mathbb{R} \]

Symmetric \( f(X) = f(V \setminus X), \quad \forall X \subseteq V. \)

Symmetric Submodular Function Minimization

\[ \min \{ f(X) \mid \phi \neq X \neq V \} ? \]

\[ O(n^3) \quad \text{Queyranne (1998)} \]
CONSTRAINED SUBMODULAR MINIMIZATION

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>hardness ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex cover</td>
<td>2</td>
<td>2 [UGC]</td>
<td>Khot,Regev ’03</td>
</tr>
<tr>
<td>$k$-unif. hitting set</td>
<td>$k$</td>
<td>$k$ [UGC]</td>
<td>Khot,Regev ’03</td>
</tr>
<tr>
<td>$k$-way partition</td>
<td>$2 - 2/k$</td>
<td>$2 - 2/k$</td>
<td>Ene,V.,Wu ’12</td>
</tr>
<tr>
<td>Facility location</td>
<td>log $n$</td>
<td>log $n$</td>
<td>Svitkina,Tardos ’07</td>
</tr>
<tr>
<td>Set cover</td>
<td>$n$</td>
<td>$n / \log^2 n$</td>
<td>Iwata,Nagano ’09</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
<td>\geq k$</td>
<td>$O(\sqrt{n})$</td>
</tr>
<tr>
<td>Sparsest Cut</td>
<td>$O(\sqrt{n})$</td>
<td>$\Omega(\sqrt{n})$</td>
<td>Svitkina,Fleischer ’09</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>$O(\sqrt{n})$</td>
<td>$\Omega(\sqrt{n})$</td>
<td>Svitkina,Fleischer ’09</td>
</tr>
<tr>
<td>Shortest path</td>
<td>$O(n^{2/3})$</td>
<td>$\Omega(n^{2/3})$</td>
<td>GKTW ’09</td>
</tr>
<tr>
<td>Spanning tree</td>
<td>$O(n)$</td>
<td>$\Omega(n)$</td>
<td>GKTW ’09</td>
</tr>
</tbody>
</table>
## MONOTONE MAXIMIZATION

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>S</td>
<td>\leq k$</td>
<td>$1 - 1/e$</td>
</tr>
<tr>
<td>matroid</td>
<td>$1 - 1/e$</td>
<td>$1 - 1/e$</td>
<td>multilinear ext.</td>
</tr>
<tr>
<td>$O(1)$ knapsacks</td>
<td>$1 - 1/e$</td>
<td>$1 - 1/e$</td>
<td>multilinear ext.</td>
</tr>
<tr>
<td>$k$ matroids</td>
<td>$k + \varepsilon$</td>
<td>$k/\log k$</td>
<td>local search</td>
</tr>
<tr>
<td>$k$ matroids &amp; $O(1)$ knapsacks</td>
<td>$O(k)$</td>
<td>$k/\log k$</td>
<td>multilinear ext.</td>
</tr>
</tbody>
</table>

## NON-MONOTONE MAXIMIZATION

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconstrained</td>
<td>$1/2$</td>
<td>$1/2$</td>
<td>combinatorial</td>
</tr>
<tr>
<td>matroid</td>
<td>$1/e$</td>
<td>$0.48$</td>
<td>multilinear ext.</td>
</tr>
<tr>
<td>$O(1)$ knapsacks</td>
<td>$1/e$</td>
<td>$0.49$</td>
<td>multilinear ext.</td>
</tr>
<tr>
<td>$k$ matroids</td>
<td>$k + O(1)$</td>
<td>$k/\log k$</td>
<td>local search</td>
</tr>
<tr>
<td>$k$ matroids &amp; $O(1)$ knapsacks</td>
<td>$O(k)$</td>
<td>$k/\log k$</td>
<td>multilinear ext.</td>
</tr>
</tbody>
</table>