Active Learning and Optimized Information Gathering

Lecture 13 – Submodularity (cont’d)

CS 101.2
Andreas Krause
Announcements

- **Homework 2**: Due Thursday Feb 19
- **Project milestone due**: Feb 24
  - 4 Pages, NIPS format: [http://nips.cc/PaperInformation/StyleFiles](http://nips.cc/PaperInformation/StyleFiles)
  - Should contain preliminary results (model, experiments, proofs, ...) as well as timeline for remaining work
  - Come to office hours to discuss projects!
- **Office hours**
  - Come to office hours before your presentation!
  - Andreas: **Monday 3pm-4:30pm**, 260 Jorgensen
  - Ryan: Wednesday 4:00-6:00pm, 109 Moore
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:

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

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

Uncertainty before knowing $X_A$  
Uncertainty after knowing $X_A$
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 $A^* = \arg\max_{A} F(A)$ with $|A| \leq k$

**NP-hard!**

**Greedy algorithm:**
- Start with $A = \emptyset$
- For $i = 1$ to $k$
  - $s^* := \arg\max_{s} F(A \cup \{s\})$
  - $A := A \cup \{s^*\}$

How well can this simple heuristic do?
Key property: Diminishing returns

Selection A = {}  
Selection B = \{X_2, X_3\}

Adding $X_1$ will help a lot!  
Adding $X_1$ doesn't help much

New feature $X_1$

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

Theorem [Krause, Guestrin UAI ’05]: Information gain $F(A)$ in Naïve Bayes models is submodular!
Why is submodularity useful?

**Theorem** [Nemhauser et al. ‘78]

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

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

~63%

- Greedy algorithm gives near-optimal solution!
- For info-gain: Guarantees best possible unless $P = NP$!
  [Krause, Guestrin UAI ’05]

Submodularity is an incredibly useful and powerful concept!
Monitoring water networks
[Krause et al, J Wat Res Mgt 2008]

- Contamination of drinking water could affect millions of people

- Place sensors to detect contaminations

- “Battle of the Water Sensor Networks” competition

Where should we place sensors to quickly detect contamination?
Model-based sensing

Utility of placing sensors based on model of the world

- For water networks: Water flow simulator from EPA
- \( F(A) = \text{Expected impact reduction placing sensors at } A \)

Model predicts Low impact

**Theorem** [Krause et al., J Wat Res Mgt '08]:

Impact reduction \( F(A) \) in water networks is submodular!

Set \( V \) of all network junctions

<table>
<thead>
<tr>
<th>Location</th>
<th>Sensor reduces impact through early detection!</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
</tr>
</tbody>
</table>

High impact reduction \( F(A) = 0.9 \)

Low impact reduction \( F(A) = 0.01 \)
Battle of the Water Sensor Networks Competition

- Real metropolitan area network (12,527 nodes)
- Water flow simulator provided by EPA
- 3.6 million contamination events
- Multiple objectives:
  - Detection time, affected population, ...
- Place sensors that detect well “on average”
What about worst-case?
[Krause et al., NIPS ’07]

Knowing the sensor locations, an adversary contaminates here!

Placement detects well on “average-case” (accidental) contamination

Very different average-case impact, Same worst-case impact

Where should we place sensors to quickly detect in the worst case?
Constrained maximization: Outline

Utility function $F(A)$

Selected set $A \subseteq Y$

Selection cost $C(A) \leq B$

Budget

Subset selection

Robust optimization

Complex constraints
Separate utility function $F_i$ for each contamination $i$

$F_i(A) = \text{impact reduction by sensors } A \text{ for contamination } i$

Want to solve:

$$A^* = \arg\max_{|A| \leq k} \min_i F_i(A)$$

Each of the $F_i$ is submodular.

Unfortunately, $\min_i F_i$ not submodular!

How can we solve this robust optimization problem?
How does the greedy algorithm do?

V = {试行, 葫芦, 苹果}
Can only buy k = 2

Optimal solution
Optimal score: 1

Hence we can’t find any approximation algorithm.
Or can we?

Greedy picks first
Then, can choose only 试行 or 葫芦
Greedy score: ε

➡️ Greedy does arbitrarily badly. Is there something better?

**Theorem:** The problem \( \max_{|A| \leq k} \min_i F_i(A) \) does not admit any approximation unless \( P = NP \)
If somebody told us the optimal value, 

$$c^* = \max_{|A| \leq k} \min_i F_i(A)$$

can we recover the optimal solution $A^*$?

Need to find

$$A^* = \arg\min_{A} |A| \text{ such that } \min_i F_i(A) \geq c^*$$

Is this any easier?

Yes, if we relax the constraint $|A| \leq k$
Solving the alternative problem

Trick: For each $F_i$ and $c$, define truncation

$$F'_{i,c}(A) = \min\{F_i(A), c\}$$

$$F'_{\text{avg},c}(A) = \frac{1}{m} \sum_i F'_{i,c}(A)$$

Remains submodular!

Problem 1 (last slide)

$$\min_{A} |A|$$

s.t. $\min_i F_i(A) \geq c$

Non-submodular ☹

Don’t know how to solve

Problem 2

$$\min_{A} |A|$$

s.t. $F'_{\text{avg},c}(A) \geq c$

Submodular!

But appears as constraint?

Same optimal solutions!
Solving one solves the other
Maximization vs. coverage

Previously: Wanted

\[ A^* = \text{argmax } F(A) \text{ s.t. } |A| \leq k \]

Now need to solve:

\[ A^* = \text{argmin } |A| \text{ s.t. } F(A) \geq Q \]

Greedy algorithm:

Start with \( A := \emptyset \);

While \( F(A) < Q \) and \( |A| < n \)

\[ s^* := \text{argmax}_s F(A \cup \{s\}) \]

\[ A := A \cup \{s^*\} \]

**Theorem** [Wolsey et al]: Greedy will return \( A_{\text{greedy}} \)

\[ |A_{\text{greedy}}| \leq (1 + \log \max_s F(\{s\})) |A_{\text{opt}}| \]

For bound, assume \( F \) is integral.
If not, just round it.
Trick: For each $F_i$ and $c$, define truncation

$$F'_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), c\}$$

$$F'_{\text{avg},c}(\mathcal{A}) = \frac{1}{m} \sum_i F'_{i,c}(\mathcal{A})$$

**Problem 1 (last slide)**

$$\min_{\mathcal{A}} |\mathcal{A}|$$

s.t. $\min_{i} F_i(\mathcal{A}) \geq c$

Non-submodular 😞

Don’t know how to solve

**Problem 2**

$$\min_{\mathcal{A}} |\mathcal{A}|$$

s.t. $F'_{\text{avg},c}(\mathcal{A}) \geq c$

Submodular!

Can use greedy algorithm!
Back to our example

- Guess $c=1$
- First pick 🎸
- Then pick 🎸
  ➔ Optimal solution!

<table>
<thead>
<tr>
<th>Set A</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$\min_i F_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎸</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>🎸</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>🍎</td>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
</tr>
</tbody>
</table>

How do we find $c$?

Do binary search!
Given: set $V$, integer $k$ and monotonic SFs $F_1, \ldots, F_m$

Initialize $c_{\text{min}} = 0$, $c_{\text{max}} = \min_i F_i(V)$

Do binary search: $c = (c_{\text{min}} + c_{\text{max}}) / 2$

- Greedily find $A_G$ such that $F'_{\text{avg},c}(A_G) = c$
- If $|A_G| \leq \alpha k$: increase $c_{\text{min}}$
- If $|A_G| > \alpha k$: decrease $c_{\text{max}}$

until convergence
Theoretical guarantees
[Krause et al, NIPS ‘07]

**Theorem:** The problem \( \max_{|A| \leq k} \min_i F_i(A) \)
does not admit any approximation unless \( P=NP \)

**Theorem:** \( SATURATE \) finds a solution \( A_S \) such that

\[
\min_i F_i(A_S) \geq OPT_k \text{ and } |A_S| \leq \alpha k
\]

where

\[
OPT_k = \max_{|A| \leq k} \min_i F_i(A)
\]

\[
\alpha = 1 + \log \max_s \sum_i F_i(\{s\})
\]

**Theorem:**
If there were a polytime algorithm with better factor \( \beta < \alpha \), then \( NP \subseteq DTIME(n^{\log \log n}) \)
Example: Lake monitoring

- Monitor pH values using robotic sensor transect

True (hidden) pH values

Prediction at unobserved locations

Use **probabilistic model** (Gaussian processes) to estimate prediction error

Where should we sense to **minimize our maximum error**?

**Robust submodular optimization problem!**

\[
\min_s \text{Var}(s) - \text{Var}(s \mid A)
\]

(often) submodular

[Das & Kempe ’08]
Comparison with state of the art

Algorithm used in geostatistics: *Simulated Annealing*

[Sacks & Schiller ’88, van Groeningen & Stein ’98, Wiens ’05,...]

7 parameters that need to be fine-tuned

*SATURATE* is competitive & 10x faster

No parameters to tune!
Results on water networks

60% lower worst-case detection time!

No decrease until all contaminations detected!
**Worst- vs. average case**

Given: Set $V$, submodular functions $F_1, \ldots, F_m$

<table>
<thead>
<tr>
<th>Average-case score</th>
<th>Worst-case score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{ac}(A) = \frac{1}{m} \sum_{i} F_i(A)$</td>
<td>$F_{wc}(A) = \min_{i} F_i(A)$</td>
</tr>
</tbody>
</table>

Want to optimize both average- and worst-case score!

Can modify $SATURATE$ to solve this problem! 😊

- **Want:** $F_{ac}(A) \geq c_{ac}$ and $F_{wc}(A) \geq c_{wc}$
- **Truncate:** $\min\{F_{ac}(A), c_{ac}\} + \min\{F_{wc}(A), c_{wc}\} \geq c_{ac} + c_{wc}$
Worst- vs. average case

Can find good compromise between average- and worst-case score!
Constrained maximization: Outline

\[ \max_{\mathcal{A} \subseteq \mathcal{Y}} F(\mathcal{A}) \]
subject to \[ C(\mathcal{A}) \leq B \]

- Utility function
- Selected set
- Selection cost
- Budget
- Subset selection
- Robust optimization
- Complex constraints
Other aspects: Complex constraints

\[ \max_A F(A) \text{ or } \max_A \min_i F_i(A) \] subject to

- So far: \[ |A| \leq k \]
- In practice, more complex constraints:
  - Different costs: \[ C(A) \leq B \]

Locations need to be connected by paths
[Chekuri & Pal, FOCS ’05]
[Singh et al, IJCAI ’07]

Sensors need to communicate (form a routing tree)

Lake monitoring

Building monitoring
Non-constant cost functions

For each \( s \in V \), let \( c(s) > 0 \) be its cost (e.g., feature acquisition costs, …)

Cost of a set \( C(A) = \sum_{s \in A} c(s) \) \((\text{modular function!})\)

Want to solve

\[
A^* = \arg\max F(A) \quad \text{s.t.} \quad C(A) \leq B
\]

Cost-benefit greedy algorithm:

Start with \( A := \emptyset \);

While there is an \( s \in V \setminus A \) \( \text{s.t.} \ C(A \cup \{s\}) \cdot B \)

\[
s^* = \arg\max_{s : C(A \cup \{s\}) \leq B} \frac{F(A \cup \{s\}) - F(A)}{c(s)}
\]

\[A := A \cup \{s^*\}\]
Performance of cost-benefit greedy

Want

$max_A F(A) \text{ s.t. } C(A) \leq 1$

<table>
<thead>
<tr>
<th>Set A</th>
<th>$F(A)$</th>
<th>$C(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${a}$</td>
<td>$2\varepsilon$</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>${b}$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Cost-benefit greedy picks $a$.
Then cannot afford $b$!

$\Rightarrow$ Cost-benefit greedy performs arbitrarily badly!
Cost-benefit optimization
[Wolsey ‘82, Sviridenko ’04, Leskovec et al ’07]

Theorem

- $A_{\text{CB}}$: cost-benefit greedy solution and
- $A_{\text{UC}}$: unit-cost greedy solution (i.e., ignore costs)

Then

$$\max \{ F(A_{\text{CB}}), F(A_{\text{UC}}) \} \geq \frac{1}{2} (1-1/e) \text{OPT}$$

Can still compute online bounds and speed up using lazy evaluations

Note: Can also get

- $(1-1/e)$ approximation in time $O(n^4)$ [Sviridenko ’04]
- Slightly better than $\frac{1}{2} (1-1/e)$ in $O(n^2)$ [Wolsey ‘82]
Example: Cascades in the Blogosphere
[Leskovec, Krause, Guestrin, Faloutsos, VanBriesen, Glance ‘07]

Which blogs should we read to learn about big cascades early?

Learn about story after us!
Water vs. Web

In both problems we are given:
- Graph with nodes (junctions / blogs) and edges (pipes / links)
- Cascades spreading dynamically over the graph (contamination / citations)

Want to pick nodes to detect big cascades early

In both applications, utility functions submodular 😊
[Generalizes Kempe et al, KDD ’03]
Performance on Blog selection

Outperforms state-of-the-art heuristics
700x speedup using submodularity!
Cost of reading a blog

- Naïve approach: Just pick 10 best blogs
- Selects big, well known blogs (Instapundit, etc.)
- These contain many posts, take long to read!

Cost-benefit optimization picks summarizer blogs!

Cost(A) = Number of posts / day

Cost/benefit analysis

Cascades captured

Cost ignoring cost
Predicting the “hot” blogs

- Want blogs that will be informative in the future
- Split data set; train on historic, test on future

Blog selection “overfits” to training data!

Poor generalization!

Let’s see what goes wrong here.

Cost(A) = Number of posts / day
Robust optimization

“Overfit” blog selection \( \mathbf{A} \)

\[ F_i(\mathbf{A}) = \text{detections in interval } i \]

Optimize worst-case

“Robust” blog selection \( \mathbf{A}^* \)

\[ \mathbf{A}^* = \arg\max \min_i F_i(\mathbf{A}) \text{ s.t. } |\mathbf{A}| \leq k \]

Robust optimization \( \Leftrightarrow \) Regularization!
Predicting the “hot” blogs

Greedy on historic
Test on future
Robust solution
Test on future

Greedy on future
Test on future
“Cheating”

Cascades captured

Cost(A) = Number of posts / day

50% better generalization!
max_A F(A) or max_A \min_i F_i(A) subject to

- So far: \( |A| \leq k \)
- In practice, more complex constraints:
  - Different costs: \( C(A) \leq B \)

Locations need to be connected by paths
[Chekuri & Pal, FOCS ’05]
[Singh et al, IJCAI ’07]

Sensors need to communicate (form a routing tree)

Lake monitoring

Building monitoring
Naïve approach: Greedy-connect

- Simple heuristic: **Greedily** optimize submodular utility function $F(A)$
- Then *add* nodes to minimize communication cost $C(A)$

Want to find optimal tradeoff between information and communication cost
The pSPIEL Algorithm
[Krause, Guestrin, Gupta, Kleinberg IPSN 2006]

- **pSPIEL**: Efficient nonmyopic algorithm

  (padded Sensor Placements at Informative and cost-Effective Locations)

- Decompose sensing region into small, well-separated clusters
- Solve cardinality constrained problem **per cluster** (greedy)
- Combine solutions using k-MST algorithm
Guarantees for \textit{pSPIEL}

[Krause, Guestrin, Gupta, Kleinberg IPSN 2006]

\textbf{Theorem:}
\textit{pSPIEL} finds a tree $T$ with

\begin{align*}
\text{submodular utility} & \quad F(T) \geq \Omega(1) \quad \text{OPT}_F \\
\text{communication cost} & \quad C(T) \leq O(\log |V|) \quad \text{OPT}_C
\end{align*}
What you should know

- Many important objective functions in Bayesian experimental design are monotonic & submodular
  - Entropy
  - Information gain*
  - Variance reduction*
  - Detection likelihood / time

- Greedy algorithm gives near-optimal solution
- Can also solve more complex problems
  - Connectedness-constraints (trees/paths)
  - Robustness

*under certain assumptions