Deep learning and continuous optimization

Spring semester 2025/26

Kristóf Bérczi

Eötvös Loránd University
Institute of Mathematics
Department of Operations Research

Midterm

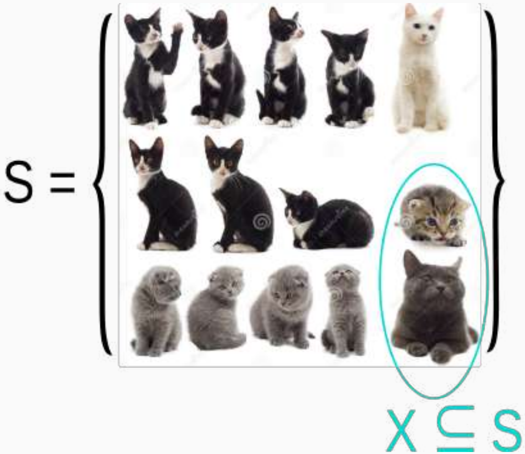
Date: Thursday, 19 March 2026

Time: 14:00-15:30

Evaluation: Max 50pts.

Lecture 5: Submodular functions

(Finite) **Ground Set:**



We consider **set functions**: $f(X) = 3$.

Motivation

Semantic segmentation:

Question: How can we map pixels to objects?

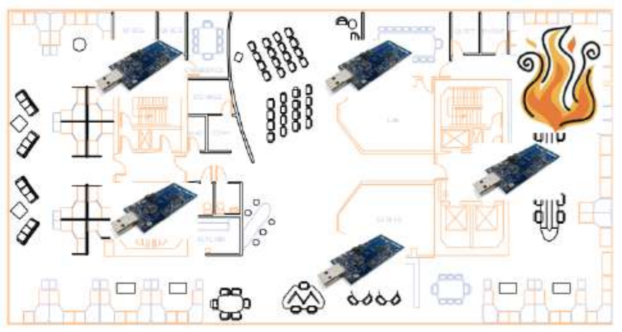
Motivation

Document summarization:

Question: How can we select representative sentences?

Motivation

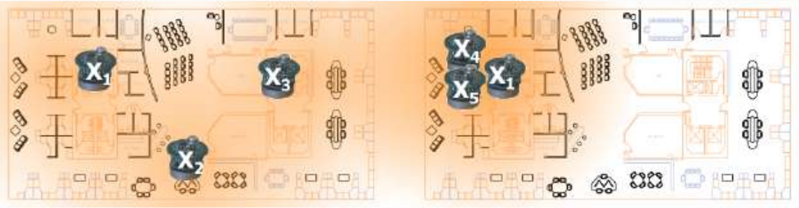
Sensor placement:



Question: How to place the sensors optimally?

Motivation

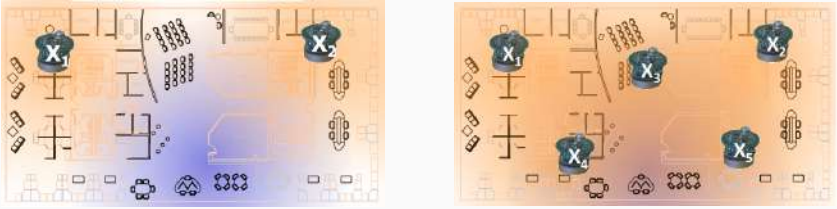
Sensor placement:



Obs. Some placements are more effective than others.

Motivation

Sensor placement:



Obs. Adding a new sensor has "more value" in the first case than in the second case.

Discrete optimization

Setup: Given a set $\mathcal{F}$ of feasible solutions and a function $f : \mathcal{F} \to \mathbb{R}$, solve

$$\max\{f(X) : X \in \mathcal{F}\}$$

$$\min\{f(X) : X \in \mathcal{F}\}$$

Arbitrary set functions are hopelessly difficult to optimize...
For 100 items, we should check 2^{100} sets!

Goal: Find sufficient conditions that make the problem tractable.

Recall: In the continuous case, $f : \mathbb{R}^n \to \mathbb{R}$ can be **minimized** if f is **convex**, and **maximized** if f is **concave**.

$\Rightarrow$ Is it possible to find discrete counterparts?

Note: Many problems in real life applications assume a discrete setting, therefore it would be crucial to provide efficient algorithms.

Set functions

Let S be a set of size n. A **set function** is a function of the form $f : 2^S \to \mathbb{R}$, where 2^S denotes the set of all subsets of S.

Given a set $X \subseteq S$ and $s \in S$, we denote by

$$X + s := X \cup \{s\},$$
$$X - s := X \setminus \{s\}.$$

The **marginal value** of s w.r.t. X is

$$f(s|X) = f(X + s) - f(X).$$

Further properties:

- **Monotone:** if $X \subseteq Y \subseteq S$, then $f(X) \leqslant f(Y)$.
- **Nonnegative:** $f(X) \geqslant 0$ for $X \subseteq S$.
- **Normalized:** $f(\emptyset) = 0$ (we will usually assume this throughout).

Modular functions

A set function $f : 2^S \to \mathbb{R}$ is **modular** if for all $X \subseteq S$ we have

$$f(X) = \sum_{s \in X} f(s).$$

Intuitively: Associate a weight w_s with each $s \in S$, and set $f(X) = \sum_{s \in X} w_s$.

e.g., $w($$) = 2$; $w($$) = 3$; $\Rightarrow f(\{$, $\}) = 5$

$\Rightarrow$ Discrete analogue of linear functions.

Submodularity

A set function $f : 2^S \to \mathbb{R}$ is **submodular** if for all $X \subseteq Y \subseteq S$ and $s \in S \setminus Y$ we have

$$f(s|X) \geqslant f(s|Y).$$

Intuitively: The gain is more from a new element if we start with a smaller set.

Example: $f(\text{new car}|\{\text{bike}\}) \geqslant f(\text{new car}|\{\text{bike,car,private jet}\})$

[The marginal value of an element exhibits *diminishing marginal returns*.]

Remarks:

- f is **supermodular** if $-f$ is submodular
- f is **modular** if and only if it is both sub- and supermodular

Equivalent definition I

A set function $f : 2^S \to \mathbb{R}$ is submodular if and only if for all $X, Y \subseteq S$ we have

$$f(X) + f(Y) \geqslant f(X \cap Y) + f(X \cup Y).$$

Proof.
$\Leftarrow$ Let $X \subseteq Y \subseteq S$ and $s \in S \setminus Y$. Then $X \cup Y = Y$ and $X \cap Y = X$, hence

$$f(s|X) = f(X + s) - f(X) \geqslant f(Y + s) - f(Y) = f(s|Y).$$

$\Rightarrow$ Assume that f is submodular, and let $X \setminus Y = \{x_1, \dots, x_k\}$. Furthermore, let $X_i := \{x_1, \dots, x_i\}$ for $i = 1, \dots, k$. Then

$$\begin{aligned}
f((X \cap Y) \cup X_1) - f(X \cap Y) &\geqslant f(Y \cup X_1) - f(Y) \\
f((X \cap Y) \cup X_2) - f((X \cap Y) \cup X_1) &\geqslant f(Y \cup X_2) - f(Y \cup X_1) \\
&\vdots \\
f((X \cap Y) \cup X_k\}) - f((X \cap Y) \cup X_{k-1}) &\geqslant f(Y \cup X_k) - f(Y \cup X_{k-1}\}) \\
\hline
f(X) - f(X \cap Y) &\geqslant f(X \cup Y) - f(Y)
\end{aligned}$$

$\square$

Equivalent definition II

A set function $f : 2^S \to \mathbb{R}$ is supermodular if and only if for all $X, Y \subseteq S$ we have

$$f(X) + f(Y) \leqslant f(X \cap Y) + f(X \cup Y).$$

A set function $f : 2^S \to \mathbb{R}$ is modular if and only if for all $X, Y \subseteq S$ we have

$$f(X) + f(Y) = f(X \cap Y) + f(X \cup Y).$$

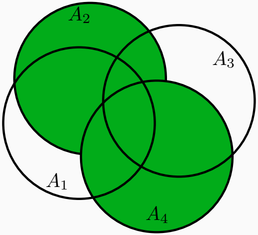
Remark: These functions play a crucial role in combinatorial optimization, and also in machine learning.

Example I - Coverage

Coverage function. Assume that for $s \in S$, we are given a measurable set A_s. Then

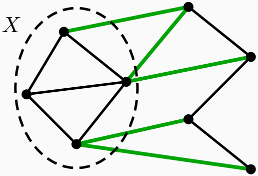
$$f(X) := \left| \bigcup_{s \in X} A_s \right|$$

is submodular.

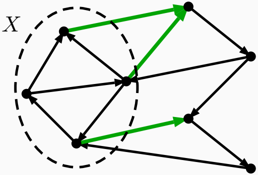


Example II - Cuts in graphs

Cut function. Let $G = (V, E)$ be an undirected graph. Then $f(X) := d_G(X)$ is submodular.



In- and out-degrees. Let $D = (V, A)$ be a directed graph. Then the out-degree $f(X) := d^+_D(X)$ and the in-degree $f(X) := d^-_D(X)$ functions are submodular.



Example III - Entropy

Entropy. Let $(\xi_s)_{s\in S}$ be random variables with finite number of values in $(\mathcal{X}_s)_{s\in S}$, respectively. For a set $X = \{s_1, \ldots, s_k\} \subseteq S$, the joint entropy is

$$f(X) = -\sum_{x_{s_1}\in\mathcal{X}_{s_1}} \cdots \sum_{x_{s_k}\in\mathcal{X}_{s_k}} P(x_{s_1}, \ldots, x_{s_k}) \log_2 P(x_{s_1}, \ldots, x_{s_k}).$$

Then f is submodular.

Mutual information. $i(X) := f(X) + f(S \setminus X) - f(S)$ is submodular.

Properties

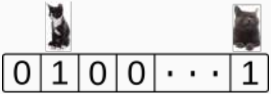
1. **Positive linear combinations:** If $f_1, \ldots, f_k$ are submodular and $\lambda_i \geqslant 0$ for $i = 1, \ldots, k$, then $\sum_{i=1}^{k} \lambda_i f_i$ is submodular.
2. **Reflection:** If f is submodular, then $g(X) := f(S \setminus X)$ is submodular.
3. **Restriction:** If $X \subseteq S$ and f is submodular, then $g(Y) := f(X \cap Y)$ is submodular.
4. **Conditioning:** If $X \subseteq S$ and f is submodular, then $g(Y) := f(X \cup Y)$ is submodular.
5. **Contraction:** If $X \subseteq S$ and f is submodular, then $g(Y) := f(X \cup Y) - f(X)$ is submodular.
6. **Maximum/minimum:** If f and g are submodular, then $\max\{f, g\}$ and $\min\{f, g\}$ are **not** necessarily submodular.

Submodularity and concavity

Given a set $X \subseteq S$, let 1_X denote its **characteristic vector**, that is,

$$(1_X)_s = \begin{cases} 1 & \text{if } s \in X, \\ 0 & \text{otherwise.} \end{cases}$$

A set function $f : 2^S \to \mathbb{R}$ can be thought of as a function defined on $\{0,1\}^S$.



Recall: A function $f : \mathbb{R} \to \mathbb{R}$ is concave if $f'(x)$ is non-increasing in x.

Now: A function $f : \{0,1\}^S \to \mathbb{R}$ is submodular if the "discrete derivative"

$$\partial_s f(x) = f(x + e_s) - f(x)$$

is non-increasing in x.

Furthermore: If a function $g : \mathbb{R}_+ \to \mathbb{R}$ is concave, then $f(X) := g(|X|)$ is submodular.

Submodularity and convexity I

Let $f : \{0,1\}^S \to \mathbb{R}$ be a set function. For a vector $c \in \mathbb{R}^S$, let $s_1, \ldots, s_n$ be an ordering of the elements S such that $c_{s_1} \geqslant \ldots \geqslant c_{s_n}$. Furthermore, let $S_i := \{s_1, \ldots, s_i\}$ for $i = 1, \ldots, n$. The **Lovász-extension** of f on c is

$$\begin{aligned}\hat{f}(c) :&= c_{s_n} f(S_n) + \sum_{i=1}^{n-1} (c_{s_i} - c_{s_{i+1}}) f(S_i) \\ &= c_{s_1} f(S_1) + \sum_{i=2}^{n} c_{s_i} (f(S_i) - f(S_{i-1})) \\ &= c_{s_1} f(S_1) + \sum_{i=2}^{n} c_{s_i} f(s_i | S_{i-1}).\end{aligned}$$

$\Rightarrow$ The sum of the marginal gains weighted by the components of c.

Submodularity and convexity II

- $\hat{f}$ is an extension of f in the sense that $\hat{f}(1_X) = f(X)$ for $X \subseteq S$.
- $\hat{f}$ is piecewise affine.
- $\hat{f}$ is **convex** if and only if f is **submodular**.
- When restricted to $[0,1]^S$, $\hat{f}$ attains its minimum at one of the vertices, that is,

$$\min_{c \in [0,1]^S} \hat{f}(c) = \min_{X \subseteq S} f(S).$$

Conclusion: Submodular functions share properties in common with both convex and concave functions. So, can we minimize/maximize them?

Submodular minimization I

Input: A submodular function $f : 2^S \to \mathbb{R}$.

Goal: Find $\arg\min_{X \subseteq S} f(X)$.

By the properties of the Lovász extension, this is equivalent to finding

$$\arg\min_{x \in [0,1]^n} \hat{f}(x).$$

Thm.

The Lovász extension $\hat{f}$ can be minimized using the Ellipsoid method in $O(n^8 \log^2 n)$ time.

Remarks:

- $O(n^6)$ algorithm (Schrijver (2000), Iwata et al. (2001), Orlin (2009)).
- Faster algorithms in special cases (cuts, flows).

Submodular minimization II

1. **Symmetric submodular functions.** The function f is symmetric if $f(X) = f(S \setminus X)$. In this case

$$2f(X) = f(X) + f(S \setminus X) \geqslant f(\emptyset) + f(S) = 2f(\emptyset) = 0,$$

hence the minimum is trivially attained at S.
$\Rightarrow$ Usually, we are interested in $\arg\min_{\emptyset \neq X \subset S} f(X)$.

Queyranne, 1998

If f is symmetric, then there is a fully combinatorial algorithm for solving $\arg\min_{\emptyset \neq X \subset S} f(X)$ in $O(n^3)$ time.

2. **Constrained setting.** A simple constraint can make submodular minimization hard, e.g., $n^{1/2}$-hardness for $\min_{X \subseteq S, |X| \geqslant k} f(S)$.
$\Rightarrow$ In such cases, one might be interested in finding approximate solutions.

Example - Clustering

Input: A set S.

Goal: Find a partition into k clusters $S_1, \ldots, S_k$ such that

$$g(S_1, \ldots, S_k) = \sum_{i=1}^{k} f(S_i)$$

is minimized, where f is a submodular function (e.g. entropy or cut function).

Observation: For $k = 2$, the function $g(X) = f(X) + f(S \setminus X)$ is symmetric and submodular, thus Queyranne's algorithm applies.

1. Let $\mathcal{P}_1 = \{S\}$.
2. For $i = 1, \ldots, k-1$:
   a. For each $S_j \in \mathcal{P}_i$, let $\mathcal{P}_i^j$ be a partition obtained by splitting S_j using Queyranne's algorithm.
   b. Set $\mathcal{P}_{i+1} = \arg\min f(\mathcal{P}_i^j)$.

Thm.

If $\mathcal{P}$ is the partition provided by the greedy splitting algorithm, then

$$f(\mathcal{P}) \leqslant \left(2 - \frac{2}{k}\right) f(\mathcal{P}_{opt}).$$

Submodular maximization

The maximization of submodular functions naturally comes up in applications.

The function is often assumed to be monotone, that is, $f(X) \leqslant f(Y)$ for $X \subseteq Y \subseteq S$.

$\Rightarrow$ When f is monotone, then the maximum is clearly attained on S.

Hence:

- Non-monotone submodular maximization (e.g. Max Cut).
- Monotone submodular maximization with constraints (e.g. $\max_{X \subseteq S, |X| \leqslant k} f(X)$).

Monotone submodular maximization

Greedy algorithm

1. Set $S_0 := \emptyset$.
2. For $i = 1, 2, \ldots, k$:
   - Pick an element s maximizing $f(s|S_{i-1})$.
   - If the marginal value is non-negative, set $S_i := S_{i-1} + s$.
   - Otherwise stop.

Nemhauser, Wolsey, Fisher

The greedy algorithm gives a $(1 - \frac{1}{e})$-approximation for the problem $\max_{X \subseteq S, |X| \leqslant k} f(X)$, where f is monotone submodular.

Remark:

- When instead of $|X| \leqslant k$ a matroid constraint $X \in \mathcal{I}$ is given, then the greedy algorithm gives a $\frac{1}{2}$-approximation.

Further approaches

1. **Partial enumeration:** Guess the first few elements, then run the greedy algorithm.
2. **Local search:** Switch up to t elements if the function value is decreased.
   - 1/3-approximation for unconstrained (non-monotone) maximization
   - Further results for matroid constraints.

Summary

	Maximization	Minimization
Unconstrained	NP-hard (Good approximations)	Polynomial time solvable (Slow, but we can exploit special cases)
constrained	NP-hard (Good approximations: some greedy-like, but greedy is not good for more complex problems)	NP-hard (hard to approximate, but still useful algorithms)