Submodular Functions and Machine Learning

MLSS Kyoto
Stefanie Jegelka
MIT
Set functions

ground set

\( \mathcal{V} = \{ \text{salad}, \text{sandwich}, \text{sandwich}, \text{apple}, \text{fries}, \text{soda} \} \)

\( F : 2^\mathcal{V} \rightarrow \mathbb{R} \)

\( F(\text{fries, soda}) = \text{cost of buying items together, or} \)

\( \text{utility, or} \)

probability, ...

We will assume:

- \( F(\emptyset) = 0 \)
- black box “oracle” to evaluate \( F \)
Discrete Labeling
Summarization

\[ F(S) = \text{relevance} + \text{diversity or coverage} \]
Informative Subsets

- where put sensors?
- which experiments?
- summarization

\[ F(S) = \text{“information”} \]
Sparsity

\[ y = A x + \text{noise} \]

\[ F(S) = \text{“penalty on support pattern”} \]
Formalization

- Formalization:
  Optimize a set function $F(S)$ (under constraints)

- generally very hard 😞
- submodularity helps:
  efficient optimization & inference with guarantees! 😊
Roadmap

• Submodular set functions
  – definition & basic properties
  – links to convexity
  – special polyhedra

• Minimizing submodular functions
  coherence, regularization, convexity

• Maximizing submodular functions
  diversity, repulsion, concavity
\mathcal{V} = \text{all possible locations}

F(S) = \text{information gained from locations in S}
Marginal gain

• Given set function \( F : 2^V \rightarrow \mathbb{R} \)

• Marginal gain:
  \[
  F(s|A) = F(A \cup \{s\}) - F(A)
  \]
Diminishing marginal gains

placement $A = \{1, 2\}$

placement $B = \{1, \ldots, 5\}$

Big gain

new sensor $s$

$A \subseteq B$

\[
F(A \cup s) - F(A) \geq F(B \cup s) - F(B)
\]
Submodularity

\[ A \subseteq B \]

\[ F(A \cup s) - F(A) \geq F(B \cup s) - F(B) \]

extra cost: one drink

extra cost: free refill 😊

diminishing marginal costs
Submodular set functions

- **Diminishing gains:** for all $A \subseteq B$

  $$F(A \cup e) - F(A) \geq F(B \cup e) - F(B)$$

- **Union-Intersection:** for all $S, T \subseteq \mathcal{V}$

  $$F(S) + F(T) - F(S \cup T) + F(S \cap T)$$
Supermodular set functions

• Submodularity: diminishing marginal gains

\[ F(A \cup e) - F(A) \geq F(B \cup e) - F(B) \]

• Supermodularity: increasing marginal gains

\[ F(A \cup e) - F(A) \leq F(B \cup e) - F(B) \]
The big picture

- Graph theory (Frank 1993)
- Electrical networks (Narayanan 1997)
- Game theory (Shapley 1970)
- Matroid theory (Whitney, 1935)
- Submodular functions
- Information theory
- Stochastic processes (Macchi 1975, Borodin 2003)
- Machine learning

Contributors:
- G. Choquet
- J. Edmonds
- L.S. Shapley
- L. Lovász
Examples

- each element $e$ has a weight $w(e)$

$$F(S) = \sum_{e \in S} w(e)$$

$A \subset B$

$$F(A \cup e) - F(A) = w(e) = F(B \cup e) - F(B) = w(e)$$

linear / modular function always submodular!
Examples

sensing:
F(S) = information gained from locations S
Example: cover

\[ F(S) = \left| \bigcup_{v \in S} \text{area}(v) \right| \]

\[ F(A \cup v) - F(A) \geq F(B \cup v) - F(B) \]
More complex model for sensing

Joint probability distribution

\[ P(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = P(Y_1, \ldots, Y_n) \cdot P(X_1, \ldots, X_n \mid Y_1, \ldots, Y_n) \]

\( Y_s \): temperature at location \( s \)

\( X_s \): sensor value at location \( s \)

\( X_s = Y_s + \text{noise} \)

Prior

Likelihood
Sensor placement

Utility of having sensors at subset $A$ of all locations

$$F(A) = H(Y) - H(Y | X_A) = I(Y; X_A)$$

Uncertainty about temperature $Y$
before sensing

Uncertainty about temperature $Y$
after sensing

$A=\{1,2,3\}$: High value $F(A)$

$A=\{1,4,5\}$: Low value $F(A)$
Information gain

\[ X_1, \ldots X_n, Y_1, \ldots, Y_m \text{ discrete random variables} \]

\[ F(A) = I(Y; X_A) = H(X_A) - H(X_A|Y) \text{ modular} \]

if all \( X_i, X_j \) conditionally independent given \( Y \)

then \( F \) is submodular!

(Exercise: complete the proof)
Entropy

\[ X_1, \ldots, X_n \] discrete random variables

\[ F(S) = H(X_S) = \text{joint entropy of variables indexed by } S \]

\[ A \subset B \]

\[ H(X_{A \cup e}) - H(X_A) = H(X_e | X_A) \]

\[ \leq H(X_e | X_B) \quad \text{“information never hurts”} \]

\[ = H(X_{B \cup e}) - H(X_B) \]

discrete entropy is submodular!
Submodularity and independence

$X_1, \ldots, X_n$ discrete random variables

$X_i, i \in S$ statistically independent

$\Leftrightarrow$ $H$ is modular/linear on $S$ \quad $H(X_S) = \sum_{e \in S} H(X_e)$

Similarly: linear independence

$\mathcal{V} = \{\text{vectors in } S \text{ linearly independent}\}$

$\Leftrightarrow$ $F$ is modular/linear on $S$: \quad $F(S) = |S|$
Maximizing Influence

\[ F(S) = \text{expected } \# \text{ infected nodes} \]

\[ F(S \cup s) - F(S) \geq F(T \cup s) - F(T) \]

(Kempe, Kleinberg & Tardos 2003)
Graph cuts

- Cut for one edge:

\[ F(S) = \sum_{u \in S, v \notin S} w_{uv} \]

\[ F(\{u\}) + F(\{v\}) \geq F(\{u, v\}) + F(\emptyset) \]

- cut of one edge is submodular!
- large graph: sum of edges

Useful property: sum of submodular functions is submodular
Types of submodular functions

- monotone increasing and integer-valued
  - rank functions
- monotone increasing
  - coverage
  - entropy
  - spread
- general (non-monotone)
  - graph cuts

\[ A \subseteq B \Rightarrow F(A) \leq F(B) \]
Sets and boolean vectors

any set function with \(|V| = n\).

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

... is a function on binary vectors!

\[ F : \{0, 1\}^n \rightarrow \mathbb{R} \]

subset selection = binary labeling!

\[ x = 1_A \]

\[
\begin{array}{c}
| & a & b & c & d \\
\hline
1 & 1 & 0 & 0 \\
\end{array}
\]
Attractive potentials

\[
\max_{\mathbf{x} \in \{0,1\}^n} \quad P(\mathbf{x} \mid \mathbf{z}) \propto \exp\left(-E(\mathbf{x}; \mathbf{z})\right)
\]

\[\Leftrightarrow \min_{\mathbf{x} \in \{0,1\}^n} \quad E(\mathbf{x}; \mathbf{z})\]
Attractive potentials

\[ E(x; z) = \sum_i E_i(x_i) + \sum_{ij} E_{ij}(x_i, x_j) \]

spatial coherence:

\[ E_{ij}(1, 0) + E_{ij}(0, 1) \geq E_{ij}(0, 0) + E_{ij}(1, 1) \]

\[ S = \{i\} \]
\[ T = \{j\} \]
\[ S \cap T = \emptyset \]
\[ S \cup T \]

\[ F(S) + F(T) \geq F(S \cup T) + F(S \cap T) \]
\[ P(S \mid \text{data}) \propto P(S) \, P(\text{data} \mid S) \]

“spread out”
Determinantal point processes

- normalized similarity matrix $K$
- sample $Y$:

$$P(S \subseteq Y) = \det(K_S)$$

$$P(e_i \in Y) = K_{ii}$$

$$P(e_i, e_j \in Y) = K_{ii}K_{jj} - K_{ij}^2$$

$$= P(e_i \in Y)P(e_j \in Y) - K_{ij}^2$$

$$F(S) = \log \det(K_S)$$ is submodular
Diversity priors

Figure 3: Results for pose estimation. The horizontal axis gives the distance threshold used to determine whether two parts are successfully matched. 95% confidence intervals are shown.

Figure 4: Structured marginals for the pose estimation task on successive steps of the sampling algorithm, with already selected poses superimposed. Input images are shown on the left. For illustration, we show the sampling process for a few images in Figure 4. As in Figure 1b, the SDPP efficiently discounts poses that are similar to those already selected.

6 Conclusion
We introduced the structured determinantal point process (SDPP), a probabilistic model over sets of structures such as sequences, trees, or graphs. We showed the intuitive “diversification” properties of the SDPP, and developed efficient message-passing algorithms to perform inference through a dual characterization of the standard DPP and a natural factorization.

Acknowledgments
The authors were partially supported by NSF Grant 0803256.

(Kulesza & Taskar 10)
Submodularity and machine learning

- Distributions over labels, sets
  - Log-submodular/supermodular probability
  - E.g. “attractive” graphical models, determinantal point processes

- (Convex) regularization
  - Submodularity: “discrete convexity”
  - E.g. combinatorial sparse estimation

- Submodularity in machine learning!
  - Diffusion processes, covering, rank, connectivity, entropy, economies of scale, summarization, ...

Submodularity has been studied extensively for both the general task of combinatorial sparse estimation and the particular submodular function maximization and the particular problem of facility location. The best approximation factor known for facility location is 0.828 that is achieved by randomized rounding. This simple method has been shown to be an efficient heuristic for both maximizing submodular functions, the problem of maximizing submodular function and the particular problem of facility location. The best approximation factor for the task of submodular functions maximization is 0.632, which is proven to have an approximation factor 0.632 with an element of maximum incremental rent solution with an element of maximum incremental reduction. Unlike the problem of minimizing submodular function, the problem of maximizing submodular function is supermodular, i.e., convexity is violated. This property potentially allows greedy algorithm to achieve higher recall compared to LBP and SA. However, if greedy algorithm is used for the task of minimizing submodular function, it turned out that the iterative greedy framework greedy algorithm showed approximately the same accuracy as LBP and SA. Moreover in contrast to LBP and SA, it turned out that the iterative greedy framework greedy algorithm showed approximately the same accuracy as LBP and SA.
Closedness properties

$F(S)$ submodular on $V$. The following are submodular:

- **Restriction:**  
  \[ F'(S) = F(S \cap W) \]
$F(S)$ submodular on $V$. The following are submodular:

- **Restriction:** $F'(S) = F(S \cap W)$

- **Conditioning:** $F'(S) = F(S \cup W)$
$F(S)$ submodular on $V$. The following are submodular:

- **Restriction:** $F'(S) = F(S \cap W)$
- **Conditioning:** $F'(S) = F(S \cup W)$
- **Reflection:** $F'(S) = F(V \setminus S)$
Submodularity ...

discrete convexity ....

... or concavity?
Concave aspects

- **submodularity:**

\[ A \subseteq B, \quad s \notin B : \]
\[ F(A \cup s) - F(A) \geq F(B \cup s) - F(B) \]

- **concavity:**

\[ a \leq b, \quad s > 0 : \]
\[ f(a + s) - f(a) \geq f(b + s) - f(b) \]
Submodularity and concavity

• suppose $g : \mathbb{N} \rightarrow \mathbb{R}$ and $F(A) = g(|A|)$

$F(A)$ submodular if and only if ... $g$ is concave
Minimum of concave functions
Minimum of submodular functions

\[ F(A) = \min\{ F_1(A), F_2(A) \} \] submodular?

\[ F(A) + F(B) \geq F(A \cup B) + F(A \cap B) \] ?

<table>
<thead>
<tr>
<th>( A \cap B )</th>
<th>( F_1(A) )</th>
<th>( F_2(A) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>{a}</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>{b}</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>{a,b}</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\( \min(F_1, F_2) \) not submodular in general!
Convex functions (Lovász, 1983)

- “occur in many models in economy, engineering and other sciences”, “often the only nontrivial property that can be stated in general”
- **preserved** under many operations and transformations: larger effective range of results
- sufficient structure for a “mathematically beautiful and practically useful theory”
- efficient **minimization**

“It is less apparent, but we claim and hope to prove to a certain extent, that a similar role is played in discrete optimization by **submodular set-functions** [...]
they share the above four properties.
Convex aspects

• convex extension
• duality results
• poly-time minimization
Maximum of submodular functions

- $F_1(A), F_2(A)$ submodular. What about $F(A) = \max\{ F_1(A), F_2(A) \}$?

$F_1(A) = h(|A|)$

$F_2(A) = g(|A|)$

$\max\{ F_1, F_2 \}$ not submodular in general!
Roadmap

- Submodular set functions
  - definition & basic properties
  - links to convexity
  - special polyhedra
- Minimizing submodular functions
- Maximizing submodular functions

... and concave aspects!
any set function
with $|V| = n$.

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

... is a function on
binary vectors!

\[ F : \{0, 1\}^n \rightarrow \mathbb{R} \]

subset selection = binary labeling!
Relaxation: idea

\[
\min_{x \in \{0,1\}^n} F(x) \quad \rightarrow \quad \min_{x \in [0,1]^n} f(x)
\]
A relaxation (extension)

have

\[ F : \{0, 1\}^n \rightarrow \mathbb{R} \]

want: extension

\[ f : \mathbb{R}^n_+ \rightarrow \mathbb{R} \]

\[ x = \sum_{i=1}^{k} \alpha_i \mathbf{1}_{S_i} \]

\[ f(x) = \sum_{i=1}^{k} \alpha_i F(S_i) \]
Examples

\[ f(x) = \sum_{i=1}^{k} \alpha_i F(S_i) \]

- truncation

\[ F(S) = \max\{|S|, 1\} \]

- cut function

\[ F(S) = \begin{cases} 
1 & \text{if } S = \{1\}, \{2\} \\
0 & \text{if } S = \emptyset, \{1, 2\} 
\end{cases} \]

\[ f(x) = 0.5 + 0.5 = \max_i x_i \]

\[ \begin{array}{c|c|c}
0.5 & 1.0 & 0.5 \\
1.0 & 1.0 & 1.0 \\
\end{array} \]

\[ 1.0 - 0.5 \]

\[ f(x) = 0.5 \cdot 0 + (1 - 0.5) \cdot 1 = |x_1 - x_2| \]

“total variation”!
Alternative characterization

\[ f(x) = \sum_{i=1}^{k} \alpha_i F(S_i) \]

if \( F \) is submodular, this is equivalent to:

\[ f(x) = \max_{y \in \mathcal{B}_F} y^\top x \]

**Theorem** *(Lovász, 1983)*

Lovász extension is convex \( \iff \) \( F \) is submodular.
Theorem (Lovász, 1983)
Lovász extension is convex ⇔ F is submodular.

⇔ If F is submodular, then f is the max of linear functions.

⇒ Let $A, B \subseteq V$.

\[
f(x) = \sum_{i=1}^{k} \alpha_i F(S_i)
\]

\[
f(1_A + 1_B) \leq f(1_A) + f(1_B)
\]

Exercise: this implies that F is submodular
Alternative characterization

\[ f(x) = \sum_{i=1}^{k} \alpha_i F(S_i) \]

if \( F \) is submodular, this is equivalent to:

\[ f(x) = \max_{y \in \mathcal{B}_F} y^\top x \]

**Theorem (Lovász, 1983)**

Lovász extension is convex \( \iff \) \( F \) is submodular.
Submodular polyhedra

submodular polyhedron:

\[ \mathcal{P}_F = \{ y \in \mathbb{R}^n \mid y(A) \leq F(A) \text{ for all } A \subseteq \mathcal{V} \} \]

Base polytope

\[ y(A) = \sum_{a \in A} y_a \]

\[ \mathcal{B}_F = \{ y \in \mathcal{P}_F \mid y(\mathcal{V}) = F(\mathcal{V}) \} \]

<table>
<thead>
<tr>
<th>A</th>
<th>F(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\emptyset</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
</tr>
<tr>
<td>{a, b}</td>
<td>0</td>
</tr>
</tbody>
</table>
Base polytopes

Base polytope \( B_F = \{ y \in \mathcal{P}_F \mid y(\mathcal{V}) = F(\mathcal{V}) \} \)

2D (2 elements)

3D (3 elements)
Base polytope

\[ \mathcal{B}_F = \left\{ y \in \mathbb{R}^n \mid \sum_{a \in S} y_a \leq F(S) \text{ for all } S \subseteq \mathcal{V} \right\} \]

\[ y(\mathcal{V}) = F(\mathcal{V}) \]

Exponentially many constraints!

\[ f(x) = \max_{y \in \mathcal{B}_F} y^\top x \]

Edmonds 1970: “magic”
compute argmax in \(O(n \log n)\)

basis for submodular minimization!
Optimization over base polytope

\[ \mathcal{B}_F = \left\{ y \in \mathbb{R}^n \mid \sum_{a \in S} y_a \leq F(S) \right\} \]

\[ y(\mathcal{V}) = F(\mathcal{V}) \]

\[ f(x) = \max_{y \in \mathcal{B}_F} y^\top x \]

Edmonds’ greedy algorithm:
1. sort
   \[ x_{\pi(1)} \geq x_{\pi(2)} \geq \ldots \geq x_{\pi(n)} \]

2. chain of sets
   \[ S_0 = \emptyset, \]
   \[ S_1 = \{ \pi(1) \} \ldots \]
   \[ S_i = \{ \pi(1), \ldots, \pi(i) \} \]

\[ y_1 = F(e_1) - 0 \]

\[ y_1 + y_2 \leq F(e_1, e_2) \]

\[ y_2 = F(\{e_1, e_2\}) - F(e_1) \]
Base polytope

\[ f(x) = \sum_{i=1}^{k} \alpha_i F(S_i) \]

Edmonds’ greedy algorithm:

1. sort
   \[ x_{\pi(1)} \geq x_{\pi(2)} \geq \ldots \geq x_{\pi(n)} \]

2. chain of sets
   \[ S_0 = \emptyset, S_i = \{\pi(1), \ldots, \pi(i)\} \]

3. assign values
   \[ y_{\pi(i)} = F(S_i) - F(S_{i-1}) \]

Remarks:

- chain of sets same as before
Re-computing our examples

\[ F(S) = \max\{|S|, 1\} \]

\[ \begin{array}{c|c|c}
| & 0.5 & 1.0 \\
\hline
1.0 & 1.0 & 0 \\
\hline
\end{array} \]

\[ x = 0.5 + 0.5 = 1.0 \]

sort: \( x_2 \geq x_1 \Rightarrow S_1 = \{2\}, S_2 = \{2, 1\} \)

\[ y_2 = F(2) = 1 \]

\[ y_1 = F(2, 1) - F(2) = 1 - 1 = 0 \]

\[ f(x) = y^\top x = 1 \cdot x_1 + 0 \cdot x_2 = \max_i x_i \]

in general: \( F(S_i) - F(S_{i-1}) > 0 \) only for \( i=1 \! \)

\[ \Rightarrow f(x) = \max_i x_i \]
Re-computing our examples

\[ x = \begin{pmatrix} 0.5 \\ 1.0 \end{pmatrix} + 0.5 \begin{pmatrix} 0 \\ 1.0 \end{pmatrix} \]

\[ F(S) = \begin{cases} 1 & \text{if } S = \{1\}, \{2\} \\ 0 & \text{if } S = \emptyset, \{1, 2\} \end{cases} \]

sort: \[ x_2 \geq x_1 \Rightarrow S_1 = \{2\}, S_2 = \{2, 1\} \]

\[ y_2 = F(2) = 1 \]

\[ y_1 = F(2, 1) - F(2) = 0 - 1 = -1 \]

\[ f(x) = y^\top x = -0.5 + 1 = |x_1 - x_2| \]
Convex relaxation

1. relaxation: convex optimization (non-smooth)

2. relaxation is exact!
   \[ \Rightarrow \text{submodular minimization in polynomial time!} \]
   
   (Grötschel, Lovász, Schrijver 1981)
Amazing base polytopes

• linear optimization = Edmonds’ greedy algorithm: each vertex determined by a permutation!

Base polytopes almost everywhere:
• cores of games (Shapley)
• information theory: achievable rates for lossless coding of correlated sources (Slepian-Wolf, Cover, Fujishige)
• matroids

\[ f(x) = \max_{y \in \mathcal{B}_F} y^\top x \]
Base polytopes and spanning trees

ground set: all edges
indicator vectors $1_T$ of all spanning trees $T$

$B = \text{convex hull of all tree indicator vectors}$
is a base polytope

What is the submodular function?
$F(S) = \text{size of largest tree within } S$
$= \max\{|T| : T \text{ is a tree and } T \subseteq S\}$
Greedy algorithm for trees

- $y$ indicator vector of a tree

- Go through edges in order of their weight.

- If edge does not complete a cycle:
  
  $$F(S_i) - F(S_{i-1}) = 1 \quad y_i = 1 \quad \text{“pick i”}$$

- If it does:
  
  $$F(S_i) - F(S_{i-1}) = 0 \quad y_i = 0 \quad \text{“don’t pick i”}$$

finds the maximum weight spanning tree (Kruskal’s algorithm)
General: Matroids

Matroid $\mathcal{M} = (\mathcal{V}, \mathcal{I})$

- ground set $\mathcal{V}$
- family of independent sets $\mathcal{I}$
Matroids (semi-formally)

S is independent ( = feasible) if ...

- $|S| \leq k$

Uniform matroid

- $S$ contains at most one element from each square

Partition matroid

- $S$ contains no cycles

Graphic matroid

Matroid properties:

- $S$ independent $\Rightarrow T \subseteq S$ also independent
Matroids

S is independent (=feasible) if ...

- Uniform matroid: \(|S| \leq k\)
- Partition matroid: \(S\) contains at most one element from each group
- Graphic matroid: \(S\) contains no cycles

- \(S\) independent \(\implies T \subseteq S\) also independent
- Exchange property: \(S, U\) independent, \(|S| > |U|\) \(\implies\) some \(e \in S\) can be added to \(U: U \cup e\) independent
General: Matroids

Matroid \( M = (\mathcal{V}, \mathcal{I}) \)

- ground set \( \mathcal{V} \)
- family of independent sets \( \mathcal{I} \)

- rank function:

\[
F(S) = \max \{ |T| : T \subseteq S \text{ and } T \in \mathcal{I} \}
\]

always submodular and increasing

- another special case: matrix rank

\[
F(S) = \text{rank}(\ldots)
\]
Convex relaxation

\[
\min_{S \subseteq \mathcal{V}} F(S) = \min_{x \in \{0,1\}^n} F(x) = \min_{x \in [0,1]^n} f(x)
\]

1. relaxation: convex optimization (non-smooth)

2. relaxation is exact!
   \(\Rightarrow\) submodular minimization in polynomial time!
   
   *(Grötschel, Lovász, Schrijver 1981)*
Minimizing the Lovász extension

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

- subgradient method
- combinatorial algorithms: dual
Subgradients

subgradient: \( g_x \in \arg\max_y y^\top x \quad \text{s.t.} \quad y \in \mathcal{B}_F \)
Projected subgradient method

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

\[
x^0 = 0
\]

\[\text{for } t = 0, \ldots \text{ do}
\]

\[\text{find } g^t \in \partial f(x^t)
\]

\[x^{t+1} = \Pi_{[0,1]^n}(x^t + \alpha_t g^t)
\]

\[g^t = \arg \min_{y \in \mathcal{B}_F} y^\top x^t
\]

greedy algorithm 😊

\[
\Pi_{[0,1]^n}(y) = \\
\arg \min_{z \in [0,1]^n} \|y - z\|^2
\]
Convergence

**Theorem**
Let $D = \sqrt{n}$ and $L = \max_{g \in B_F} \|g\| \leq 3 \max_S |F(S)|$. With step size $\alpha_t = \frac{D}{L\sqrt{t}}$, the error decreases as

$$\min_{\tau \leq t} f(x^{\tau}) - f(x^*) \leq \frac{4DL}{\sqrt{t}}$$

- **D:** diameter of $[0,1]^n$
- **L:** Lipschitz constant
- for error $\leq \epsilon$ need $O\left(\frac{1}{\epsilon^2}\right)$ iterations
Submodular minimization

**convex optimization**
- ellipsoid method
  (Grötschel-Lovasz-Schrijver 81)
- subgradient method

**combinatorial methods**
- first polynomial-time:
  (Schrijver 00, Iwata-Fleischer-Fujishige 01)
- currently fastest:
  
  \[ O(n^4 T + n^5 \log M) \quad \text{(Iwata 03)} \]
  \[ O(n^6 + n^5 T) \quad \text{(Orlin 09)} \]

Latest result:

\[ O(n^2 T \log nM + n^3 \log^c nM) \]
\[ O(n^3 T \log^2 n + n^4 \log^c n) \quad \text{(Lee-Sidford-Wong 15)} \]
Convex duality

\[ \min_{S \subseteq \mathcal{V}} F(S) = \min_{x \in [0,1]^n} f(x) \]

\[ = \min_{x \in [0,1]^n} \max_{y \in \mathcal{B}_F} y^\top x \]

\[ = \max_{y \in \mathcal{B}_F} \min_{x \in [0,1]^n} x^\top y = \max_{y \in \mathcal{B}_F} \left( \sum_{i=1}^{n} \min\{y_i, 0\} \right) \]

Optimality conditions: \((S^*, y^*)\) optimal primal-dual pair if

1. \( y^* \in \mathcal{B}_F \)

2. \( \{y^* < 0\} \subseteq S^* \subseteq \{y^* \leq 0\} \)

3. \( y^*(S^*) = F(S^*) \)
Combinatorial algorithms

• solve
\[
\max_{y \in B_F} \left( \sum_{i=1}^{n} \min\{y_i, 0\} \right)
\]

• remove “negative mass”

• challenges:
  – need to stay in polytope:
  \[
  \sum_{i \in S} y_i \leq F(S)
  \]
  – cannot test feasibility

→ network flow algorithms
Submodular minimization

convex optimization

• ellipsoid method
  \((Grötschel-Lovasz-Schrijver 81)\)

• subgradient method

• minimum-norm point / Fujishige-Wolfe algorithm

combinatorial methods

• first polynomial-time:
  \((Schrijver 00, Iwata-Fleischer-Fujishige 01)\)

• currently fastest:
  \(O(n^4T + n^5 \log M)\) \((Iwata 03)\)
  \(O(n^6 + n^5T)\) \((Orlin 09)\)

Latest result:

\(O(n^2T \log nM + n^3 \log^c nM)\)  
\(O(n^3T \log^2 n + n^4 \log^c n)\) \((Lee-Sidford-Wong 15)\)
Some fun 😊

1. Complete the proof that Lovasz extension convex $\Rightarrow$ set function is submodular (slide 51)

2. Submodular oder not? Let $F$ be increasing and submodular, and define

$$G(S) = \min\{ F(S), \ c \}$$

for a constant $c$.

slides and pointers to literature:
people.csail.mit.edu/stefje/mlss
Example: costs

breakfast??

cost: time to shop + price of items

\[ F(\text{breakfast}) = \text{cost(\text{coffee})} + \text{cost(\text{fruit}, \text{sandwich})} \]

\[ = t_1 + 1 + t_2 + 2 \]

\[ = \#\text{shops} + \#\text{items} \]

submodular?
Example: costs

breakfast??

cost:
time to reach shop + price of items

ground set \( V \)

each item 1 $
Shared fixed costs

\[ F(b \mid A) = 1 + t_3 \]
\[ F(b \mid B) = 1 \]

marginal cost: \#new shops + \#new items

decreasing \Rightarrow cost is submodular!

- shops: shared fixed cost
- economies of scale