Submodular Function Optimization - An Overview

Senanayak Sesh Kumar Karri

Advisor: Prof. Francis Bach
INRIA Rocquencourt - Sierra project-team
Laboratoire d'Informatique de l'Ecole Normale Supérieure
Paris, France.
sesh-kumar.karri@inria.fr

April 23, 2013

Submodularity

Definition (submodular function)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

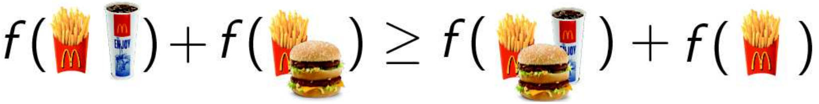
$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

Submodularity

Definition (submodular function)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

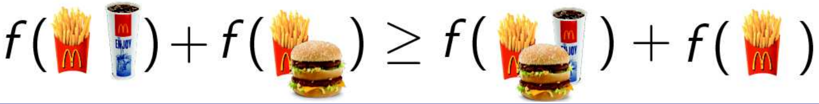


Submodularity

Definition (submodular function)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$



Definition (diminishing returns)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A \subseteq B \subset V$, and $v \in V \setminus B$, we have that:

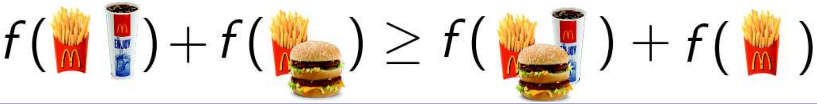
$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$$

Submodularity

Definition (submodular function)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

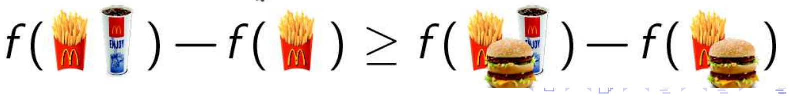
$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$



Definition (diminishing returns)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A \subseteq B \subset V$, and $v \in V \setminus B$, we have that:

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$$



Notations

- $f : 2^V \to \mathbb{R}$, where $V = \{1, 2, \ldots, p\}$. Therefore, $|V| = p$.

Notations

- $f : 2^V \to \mathbb{R}$, where $V = \{1, 2, \ldots, p\}$. Therefore, $|V| = p$.
- P : Decision problems that can be *solved* on a deterministic sequential machine(read it as "present day computer") in an amount of time that is polynomial in the size of input. Eg: To check if a number is prime etc.

Notations

- $f : 2^V \to \mathbb{R}$, where $V = \{1, 2, \ldots, p\}$. Therefore, $|V| = p$.
- P : Decision problems that can be *solved* on a deterministic sequential machine(read it as “present day computer”) in an amount of time that is polynomial in the size of input. Eg: To check if a number is prime etc.
- NP: Decision problems whose solutions can be *verified* on a deterministic sequential machine(read it as “present day computer”) in an amount of time that is polynomial in the size of input. Eg:Travelling Salesman problem. Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Notations

- $f : 2^V \to \mathbb{R}$, where $V = \{1, 2, \ldots, p\}$. Therefore, $|V| = p$.
- P : Decision problems that can be *solved* on a deterministic sequential machine(read it as "present day computer") in an amount of time that is polynomial in the size of input. Eg: To check if a number is prime etc.
- NP: Decision problems whose solutions can be *verified* on a deterministic sequential machine(read it as "present day computer") in an amount of time that is polynomial in the size of input. Eg:Travelling Salesman problem. Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- ρ-Approximate algorithms:
  - $OPT \leq f(x) \leq \rho OPT$, if $\rho > 1$.
  - $\rho OPT \leq f(x) \leq OPT$, if $\rho < 1$.

Graph Cuts

- MINIMUM CUT : Given a graph $G = (V, E)$, find a set of vertices $S \subseteq V$ that minimizes the cut (set of edges) between S and $V \setminus S$.
- MAXIMUM CUT : Given a graph $G = (V, E)$, find a set of vertices $S \subseteq V$ that minimizes the cut (set of edges) between S and $V \setminus S$.
- Weighted versions.
- Eg :- Segmentation in Computer Vision.

Image Segmentation

- An image needing to be segmented.

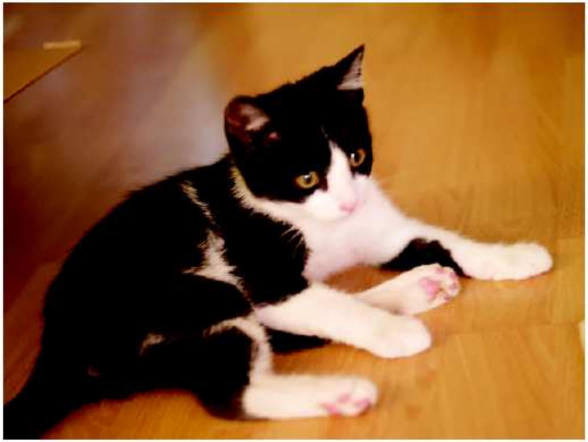


Image Segmentation

- User marks foreground(red) and background(blue).

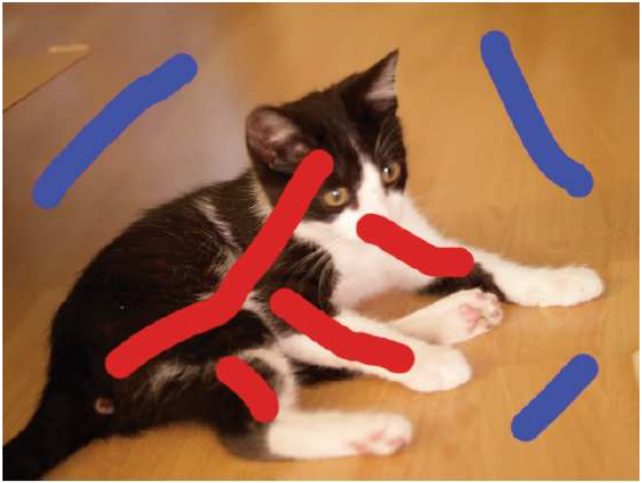
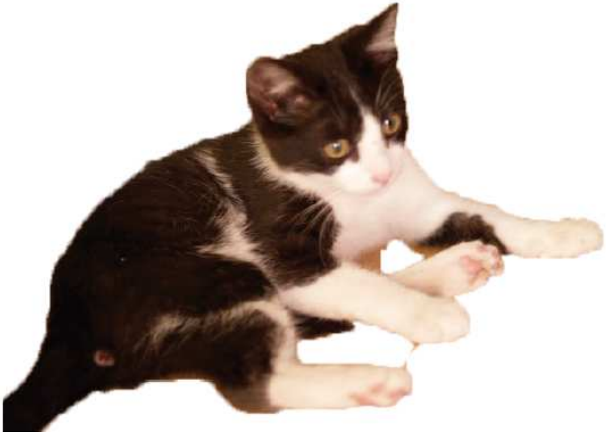


Image Segmentation

- Goal.

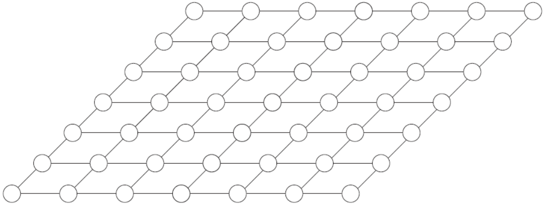


Markov random fields and image segmentation

Markov random field

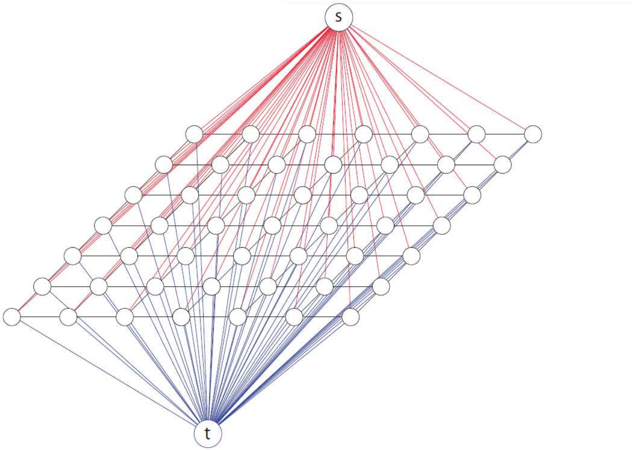
$$\log p(x) \propto \sum_{v \in V(G)} e_v(x_v) + \sum_{(i,j) \in E(G)} e_{ij}(x_i, x_j)$$

where G is a 2D grid graph, we have



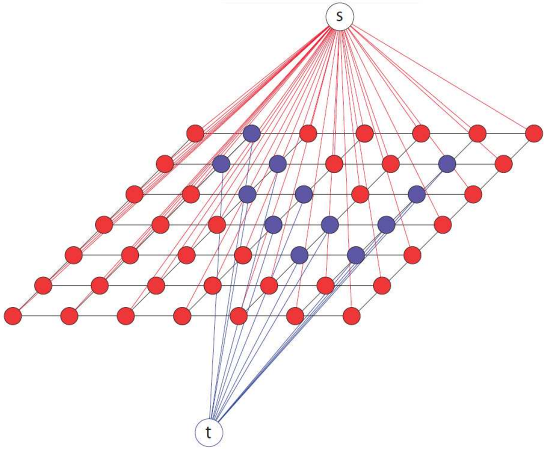
Markov random fields and image segmentation

Augmented graph-cut graph. The edge weights of graph are derived from $\{e_v\}_{v \in V}$ and $\{e_{ij}\}_{(i,j) \in E(G)}$.



Markov random fields and image segmentation

Augmented graph-cut graph with indicated cut corresponding to particular vector $\bar{x} \in \{0,1\}^n$. Each cut $\bar{x}$ has a score corresponding to $p(\bar{x})$.



Sensor Placement

- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).

Sensor Placement

- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).
- We have a function $f(S)$ that measures the “coverage” of any given set S of sensor placement decisions. Then $f(V)$ is maximum possible coverage.

Sensor Placement

- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).
- We have a function $f(S)$ that measures the "coverage" of any given set S of sensor placement decisions. Then $f(V)$ is maximum possible coverage.
- One possible goal: choose smallest set S such that $f(S) = \alpha f(V)$ with $0 < \alpha \leq 1$.

Sensor Placement

- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).
- We have a function $f(S)$ that measures the "coverage" of any given set S of sensor placement decisions. Then $f(V)$ is maximum possible coverage.
- One possible goal: choose smallest set S such that $f(S) = \alpha f(V)$ with $0 < \alpha \leq 1$.
- Another possible goal: choose size at most k set S such that $f(S)$ is maximized.

Sensor Placement

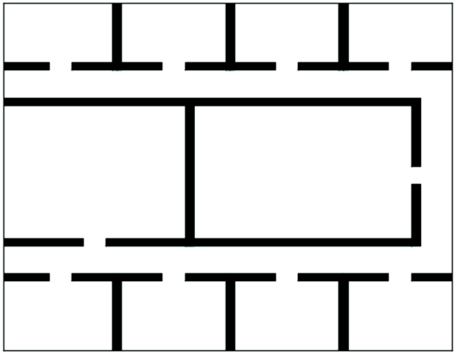
- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).
- We have a function $f(S)$ that measures the "coverage" of any given set S of sensor placement decisions. Then $f(V)$ is maximum possible coverage.
- One possible goal: choose smallest set S such that $f(S) = \alpha f(V)$ with $0 < \alpha \leq 1$.
- Another possible goal: choose size at most k set S such that $f(S)$ is maximized.
- Environment could be a floor of a building, water network, monitored ecological preservation.

Sensor Placement

- Given an environment, there is a set V of candidate locations for placement of a sensor (e.g., temperature, gas, audio, video, bacteria or other environmental contaminant, etc.).
- We have a function $f(S)$ that measures the "coverage" of any given set S of sensor placement decisions. Then $f(V)$ is maximum possible coverage.
- One possible goal: choose smallest set S such that $f(S) = \alpha f(V)$ with $0 < \alpha \leq 1$.
- Another possible goal: choose size at most k set S such that $f(S)$ is maximized.
- Environment could be a floor of a building, water network, monitored ecological preservation.

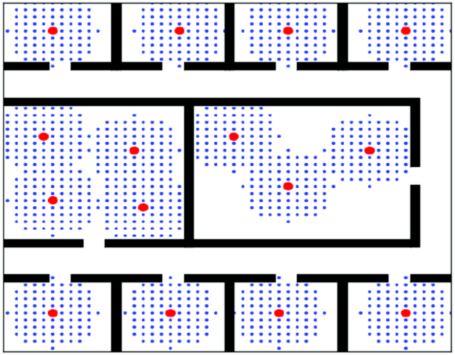
Sensor Placement in Buildings

- An example of a room layout.



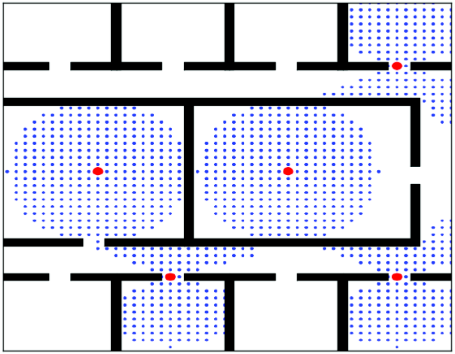
Sensor Placement in Buildings

- Small range sensors.



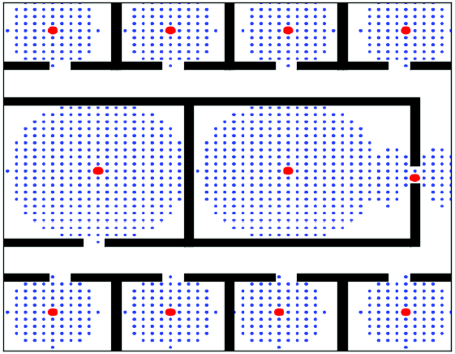
Sensor Placement in Buildings

- Large range sensors.



Sensor Placement in Buildings

- Sensors with mixed ranges.



A model of Influence in Social Networks

- Given a graph $G = (V, E)$, each $v \in V$ corresponds to a person, to each v we have an activation function $f_v : 2^V \rightarrow [0, 1]$ dependent only on its neighbours, i.e, $f_v(A) = f_v(A \cup \Gamma(v))$.

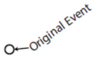
A model of Influence in Social Networks

- Given a graph $G = (V, E)$, each $v \in V$ corresponds to a person, to each v we have an activation function $f_v : 2^V \to [0, 1]$ dependent only on its neighbours, i.e, $f_v(A) = f_v(A \cup \Gamma(v))$.
- Goal - Viral Marketing: find a small subset $S \subseteq V$ of individuals to direct influence, and thus indirectly influence the greatest number of possible other individuals (via the social network G).

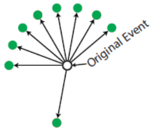
A model of Influence in Social Networks

- Given a graph $G = (V, E)$, each $v \in V$ corresponds to a person, to each v we have an activation function $f_v : 2^V \to [0, 1]$ dependent only on its neighbours, i.e, $f_v(A) = f_v(A \cup \Gamma(v))$.
- Goal - Viral Marketing: find a small subset $S \subseteq V$ of individuals to direct influence, and thus indirectly influence the greatest number of possible other individuals (via the social network G).
- We define a function $f : 2^V \to \mathbb{Z}^+$ that models the ultimate influence of an initial set S of nodes based on the following iterative process: At each step, a given set of nodes S are activated, and we activate a new node $v \in V \setminus S$ if $f_v(S) \geq U[0, 1]$(where $U[0, 1]$ is a uniform random number between 0 and 1).

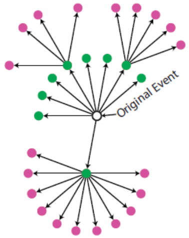
Modeling Information Cascade



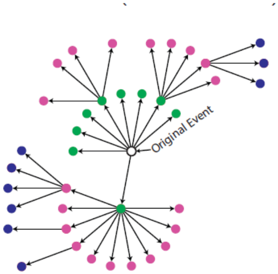
Modeling Information Cascade



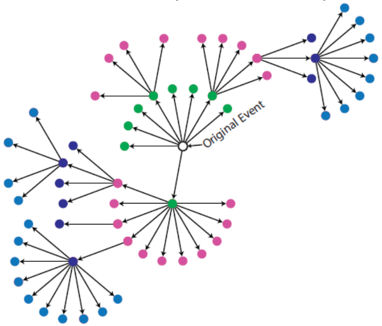
Modeling Information Cascade



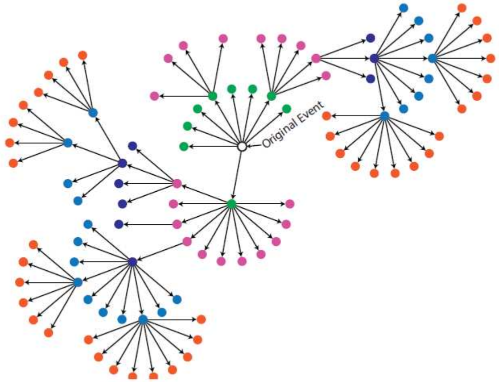
Modeling Information Cascade



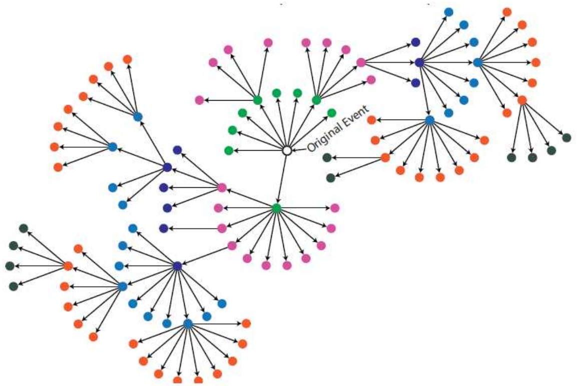
Modeling Information Cascade



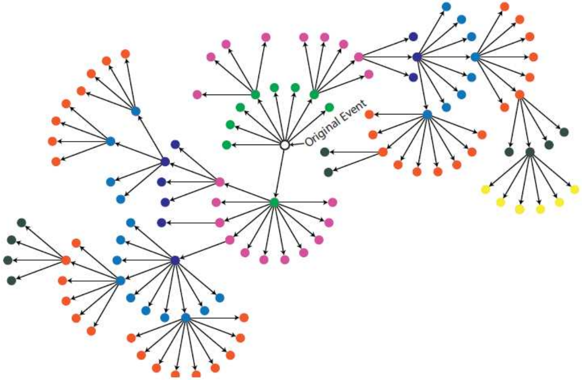
Modeling Information Cascade



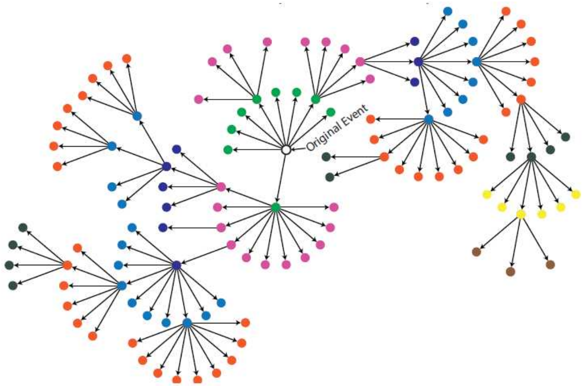
Modeling Information Cascade



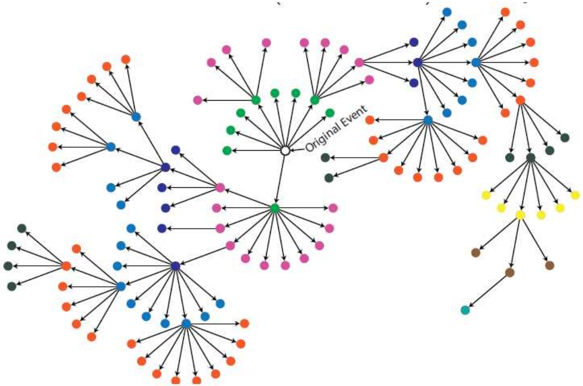
Modeling Information Cascade



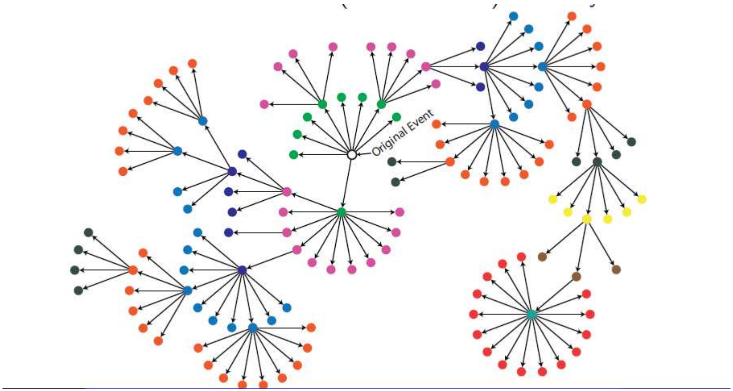
Modeling Information Cascade



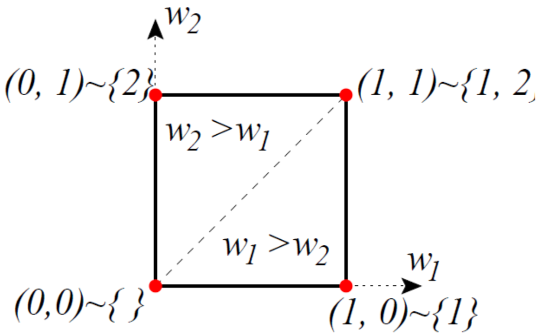
Modeling Information Cascade



Modeling Information Cascade



Submodular Minimization - Lovász extension



- Given any set function f and w such that $w_{j_1} \geq \ldots \geq w_{j_p}$, define:

$$\begin{aligned}\hat{f}(w) &= \sum_{k=1}^{p} w_{j_k}[f(\{j_1,\ldots,j_k\})] - f(\{j_1,\ldots,j_{k-1}\})] \\ &= \sum^{p-1}(w_{j_k} - w_{j_{k+1}})[f(\{j_1,\ldots,j_k\})] + w_{j_p} f(\{j_1,\ldots,j_p\})]\end{aligned}$$

Submodular Minimization

- if $w = 1_A, \hat{f}(w) = f(A) \implies$ extension from $\{0,1\}^p$ to $\mathbb{R}^p$
- $\hat{f}$ is piecewise affine and positively homogeneous
- f is submodular if and only if $\hat{f}$ is convex.
  - Minimizing $\hat{f}(w)$ on $w \in [0,1]^p$ is equivalent to minimizing f on 2^V.
  - $\min_{A \subset V} f(A) = \min_{w \in [0,1]^p} \hat{f}(w)$.

Submodular Minimization

- if $w = 1_A, \hat{f}(w) = f(A) \implies$ extension from $\{0,1\}^p$ to $\mathbb{R}^p$
- $\hat{f}$ is piecewise affine and positively homogeneous
- f is submodular if and only if $\hat{f}$ is convex.
  - Minimizing $\hat{f}(w)$ on $w \in [0,1]^p$ is equivalent to minimizing f on 2^V.
  - $\min_{A \subset V} f(A) = \min_{w \in [0,1]^p} \hat{f}(w)$.
- Exact submodular function minimization : Combinatorial algorithms.
  - Algorithms based on $\min_{A \subset V} f(A)$.
  - Best algorithms have polynomial complexity (typically $O(p^6)$ or more, where $|V| = p$).

Submodular Minimization

- if $w = 1_A, \hat{f}(w) = f(A) \implies$ extension from $\{0,1\}^p$ to $\mathbb{R}^p$
- $\hat{f}$ is piecewise affine and positively homogeneous
- f is submodular if and only if $\hat{f}$ is convex.
    - Minimizing $\hat{f}(w)$ on $w \in [0,1]^p$ is equivalent to minimizing f on 2^V.
    - $\min_{A \subset V} f(A) = \min_{w \in [0,1]^p} \hat{f}(w)$.
- Exact submodular function minimization : Combinatorial algorithms.
    - Algorithms based on $\min_{A \subset V} f(A)$.
    - Best algorithms have polynomial complexity (typically $O(p^6)$ or more, where $|V| = p$).
- Minimizing symmetric submodular functions.
    - A submodular function f is said to be symmetric if for all $B \subset V$, $f(V \setminus B) = f(B)$.
    - Example: undirected cuts, mutual information
    - Minimization in $O(p^3)$ over all non-trivial subsets of V, where $|V| = p$.

Submodular Maximization

- NP-hard to solve.

Submodular Maximization

- NP-hard to solve.
- Unconstrained Maximization.
  - Algorithms based on $\max_{A \subset V} f(A)$.
  - Feige et al.(2007) shows that for non-negative functions, a *random set* already achieves at least $1/4$ of the optimal value, while *local search* techniques achieve at least $1/2$.

Submodular Maximization

- NP-hard to solve.
- Unconstrained Maximization.
  - Algorithms based on $\max_{A \subset V} f(A)$.
  - Feige et al.(2007) shows that for non-negative functions, a *random set* already achieves at least $1/4$ of the optimal value, while *local search* techniques achieve at least $1/2$.
- Maximizing non-decreasing submodular functions with cardinality constraint
  - A submodular function f is said to be non-decreasing if for all $A \subseteq B$, $f(A) \leq f(B)$.
  - $\max_{\substack{A \subset V \\ |A| \leq k}} f(A)$.
  - Greedy algorithm achieves $(1 - 1/e)$ of the optimal value.(Nemhauser et al., 1978).

Maximization with cardinality constraint

- Let $A^* = \{b_1, \dots, b_k\}$ be a maximizer of F with k elements, and a_j the j-th selected element. Let $\rho_j = F(\{a_1, \dots, a_j\}) - F(\{a_1, \dots, a_{j-1}\})$

$$\begin{aligned} f(A^*) &\leq f(A^* \cup A_{j-1}) \text{because } f \text{ is non-decreasing,} \\ &= f(A_{j-1}) + \sum_{i=1}^{k} [f(A_{j-1} \cup \{b_1, \dots, b_i\}) \\ &\qquad - f(A_{j-1} \cup \{b_1, \dots, b_{i-1}\})] \\ &\leq f(A_{j-1}) + \sum_{i=1}^{k} [f(A_{j-1} \cup \{b_i\} - f(A_{j-1})] \text{by submodularity,} \\ &\leq f(A_{j-1}) + k\rho_j \text{ by definition of the greedy algorithm,} \\ &= \sum_{i=1}^{j-1} \rho_i + k\rho_j \end{aligned}$$

- Minimize $\sum_{i=1}^{k} \rho_i : \rho_j = (k-1)^{j-1} k^{-j} f(A^*)$

Courtesy and References

- Course on "Submodular Functions" by Prof. Jeff Bilmes, University of Washington
    - *http://j.ee.washington.edu/ bilmes/classes/ee596a_fall_2012/*
- "Learning with Submodular Functions: A Convex Optimization Perspective" by Prof. Francis Bach, INRIA.
    - *http://hal.archives-ouvertes.fr/docs/00/64/52/71/PDF/submodular_fot.pdf*
    - *http://www.di.ens.fr/ fbach/submodular_fbach_mlss2012.pdf*
- Tutorials by Prof. Andreas Krause, ETH, Zurich.
    - *http: submodularity.org*

Thank You. Questions?