Aalto University
School of Science

CS-E4070 — Computational learning theory

Slide set 10 : submodular functions II

Cigdem Aslay and Aris Gionis

Aalto University

spring 2019

submodular (set) functions

- a ground set U with n elements
- a function $f : 2^U \to \mathbb{R}$ is submodular if satisfies

  the "diminishing returns" property:

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

  for all $A \subseteq B \subseteq U$ and $x \in U \setminus B$

submodular function optimization

computational hardness differs w.r.t. the following:

- non-negativity of f : $f(A) \geq 0$ for all $A \subseteq U$
- monotonicity of f : $f(A) \leq f(B)$ for all $A \subseteq B \subseteq U$
- symmetry of f : $f(A) = f(U \setminus A)$ for all $A \subseteq U$
- constraints : cardinality, knapsack, matroid ...
- objective : maximization or minimization

submodular function maximization

monotone submodular functions

- unconstrained case: trivial
- constrained case: **NP**-hard but well-approximable

  e.g., MAX-k-COVER

monotone submodular function maximization

cardinality constraints

- find $S \subseteq U$ subject to $|S| \leq k$ that maximizes $f(S)$

- MAX k-COVER is a special case
- greedy gives $(1 - 1/e)$ approximation
  [Nemhauser et al., 1978]
- no better approximation unless **P**=**NP**

monotone submodular function maximization

cardinality constraints

1. $S \leftarrow \emptyset$
2. while $|S| < k$
3. $\quad i \leftarrow \arg\max_j f(S \cup \{j\}) - f(S)$
4. $\quad S \leftarrow S \cup \{i\}$
5. return S

monotone submodular function maximization

analysis of the greedy

- S^* : the optimal solution
- $S_j = \{x_1, \cdots, x_j\}$: the first j elements picked by the greedy
- let $f(x_j \mid S_{j-1})$ denote the marginal gain of adding the j-th element to S_{j-1}

$$f(x_j \mid S_{j-1}) = f(S_j) - f(S_{j-1})$$

- hence

$$f(S) = \sum_{j=1}^{k} f(x_j \mid S_{j-1})$$

monotone submodular function maximization

analysis of the greedy

- claim:

$$f(x_j \mid S_{j-1}) \geq \frac{f(S^*) - f(S_{j-1})}{k}$$

- proof. first we need to state a property of submodular functions:

  – if f is submodular, then the following holds $\forall A, B \subseteq U$:

$$f(A) \leq f(B) + \sum_{x \in A \setminus B} f(x \mid B) - \sum_{x \in B \setminus A} f(x \mid A \cup B \setminus \{x\})$$

  (see Proposition 2.1 in [Nemhauser et al., 1978] for all similar properties)

monotone submodular function maximization

analysis of the greedy

- proof (cont'd). using this property, we have

$$f(S^*) \leq f(S_{j-1}) + \sum_{x \in S^* \setminus S_{j-1}} f(x \mid S_{j-1}) - \sum_{x \in S_{j-1} \setminus S^*} f(x \mid S^* \cup S_{j-1} \setminus \{x\})$$

  which further implies (due to monotonicity of f):

$$f(S^*) - f(S_{j-1}) \leq \sum_{x \in S^* \setminus S_{j-1}} f(x \mid S_{j-1})$$

monotone submodular function maximization

analysis of the greedy

- proof (cont'd). using also the fact that $\forall x \in V \setminus S_{j-1}$:

  $$f(x_j \mid S_{j-1}) \geq f(x \mid S_{j-1})$$

  since otherwise x_j wouldn't be selected by greedy, we have:

  $$\begin{aligned} f(S^*) - f(S_{j-1}) &\leq \sum_{x \in S^* \setminus S_{j-1}} f(x \mid S_{j-1}) \\ &\leq k \cdot f(x_j \mid S_{j-1}) \end{aligned}$$

- we have just proved our claim

monotone submodular function maximization

analysis of the greedy

- continuing the analysis of greedy, we have

$f(S^*) - f(S_j) \le (1 - 1/k)^j f(S^*)$ (by induction)

$f(S^*) - f(S_k) \le (1 - 1/k)^k f(S^*)$

$$\begin{aligned} f(S_k) &\ge (1 - (1 - 1/k)^k) f(S^*) \\ &\ge \left(1 - \frac{1}{e}\right) f(S^*) \end{aligned}$$

monotone submodular maximization

example - max-sum diversification [Borodin et al., 2012]

- U is a ground set
- $d : U \times U \to \mathbb{R}$ is a metric distance function on U
- $f : 2^U \to \mathbb{R}$ is a submodular function

- we want to find $S \subseteq U$ such that

  $\phi(S) = f(S) + \lambda \sum_{u,v \in S} d(u, v)$ is maximized and

  $|S| \leq k$

monotone submodular maximization

example - max-sum diversification [Borodin et al., 2012]

- consider $S \subseteq U$ and $x \in U \setminus S$
- define the following types of marginal gain

  $d_x(S) = \sum_{v \in S} d(x, v)$

  $f_x(S) = f(S \cup \{x\}) - f(S)$

  $\phi_x(S) = \frac{1}{2} f_x(S) + \lambda d_x(S)$

- greedy algorithm on marginal gain $\phi_x(S)$ gives factor 2 approximation

monotone submodular function maximization

combinatorial constraints

- matroids: abstract notion of feasibility
- a matroid $M = (U, \mathcal{F})$ is a set system where U is the ground set and $\mathcal{F}$ is family of independent (feasible) subsets of U satisfying the following axioms:
  - if $A \in \mathcal{F}$ and $B \subseteq A$ then $B \in \mathcal{F}$ (downward closure)
  - if $A, B \in \mathcal{F}$ and $|B| < |A|$ then $\exists x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{F}$ (augmentation)

monotone submodular function maximization

combinatorial constraints

- uniform matroid: $A \subseteq U$ is independent if $|A| \leq k$
- partition matroid: U is partitioned in ℓ different non-empty disjoint subsets

$$U = \bigcup_{i=1}^{\ell} U_i \text{ and } U_i \cap U_j = \emptyset, \forall i, j : i \neq j$$

  - cardinality constraint k_i on each partition $U_i, \forall i \in [1, \ell]$
  - $A \subseteq U$ is independent if

$$A \cap U_i \leq k_i, \forall i \in [1, k]$$

monotone submodular function maximization

combinatorial constraints

- graphic matroid: given a graph $G = (V, E)$, define the edge set E as the ground set
- then an edge set $A \subseteq E$ is independent if the edge-induced

  graph $G_A = (V_A, E_A)$ does not contain any cycle
- $\mathcal{F}$ contains all forests and trees naturally

monotone submodular function maximization

combinatorial constraints

- given submodular monotone $f : 2^U \to \mathbb{R}_+$ and matroid constraint $M = (U, \mathcal{F})$

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

- greedy gives $(1/2)$ approximation
- in general, greedy gives $1/(1+p)$ approximation when there are p matroid constraints
  [Fisher et al., 1978]

monotone submodular function maximization

combinatorial constraints

1. $A \leftarrow \emptyset$
2. while $\exists x \in U : A \cup \{x\} \in \mathcal{F}$
3. $\quad x^* \leftarrow \arg\max_{A \cup \{x\} \in \mathcal{F}} f(A \cup \{x\}) - f(A)$
4. $\quad A \leftarrow A \cup \{x^*\}$
5. $\quad U \leftarrow U \setminus \{x^*\}$
6. return A

submodular function maximization

non-monotone submodular functions

- unconstrained case: **NP**-hard but well-approximable

  e.g., MAX-CUT

- constrained case: **NP**-hard but well-approximable

  e.g., document summarization [Lin et al., 2009]

non-monotone submodular maximization

unconstrained case

- first constant-factor approximations for non-negative submodular functions by [Feige et al., 2011]
- simple algorithms: randomized / deterministic, non-adaptive / adaptive
- $1/2$ approx for symmetric functions
- $2/5 = 0.4$ approx for the non-negative functions
- lower bound: better than $1/2$ approx requires exponential number of value queries

non-monotone submodular maximization

unconstrained case [Feige et al., 2011]

- pick a random set

  $1/4$ for non-negative function (on expectation)

  $1/2$ for symmetric function (on expectation)

- local search
  - initialize S to best singleton
  - S = local optimum (add or delete elements)
  - return the best of S and $U \setminus S$

  $1/3$ approx for non-negative function

  $1/2$ for non-negative symmetric function
- (proofs in submodularity slides - part I)

non-monotone submodular maximization

example - document summarization [Lin et al., 2009]

- U is a ground set
- $w : U \times U \to \mathbb{R}_{\geq 0}$ is a similarity function
- $f : 2^U \to \mathbb{R}$ is a submodular function

- we want to find $S \subseteq U$ such that

$$f(S) = \sum_{i \in U \setminus S} \sum_{j \in S} w(i,j) - \lambda \sum_{i,j \in S : i \neq j} w(i,j)$$

is maximized and $|S| \leq k$

submodular function minimization

- unconstrained case: polynomial-time

  e.g. MIN-CUT

- constrained case: **NP**-hard and (mostly) hard to approximate

  e.g., set cover

concave or convex

- argument for concavity: behavior looks more like concavity

  i.e., discrete derivative

  $$f(A \cup \{x\}) - f(A)$$

  is non-increasing in x

- argument for convexity: minimization problem seems to benefit more from submodularity (polynomial-time unconstrained minimization)

set functions are pseudo-Boolean functions

- any set $A \subseteq U$ can be represented as a binary vector
- the characteristic vector of a set A is given by $\mathbf{1_A} \in \{0,1\}^U$ where $\forall u \in U$

$$\mathbf{1_A}(u) = \begin{cases} 1 & \text{if } u \in A \\ 0 & \text{otherwise} \end{cases}$$

- we will use $f : \{0,1\}^U \to \mathbb{R}$ and $f : 2^U \to \mathbb{R}$ interchangeably

the Lovász extension

- given $f : \{0,1\}^U \to \mathbb{R}$, its Lovász extension is the function $f^L : [0,1]^U \to \mathbb{R}$ defined as

$$f^L(\mathbf{x}) = \sum_{i=0}^{n} \alpha_i f(A_i)$$

where $\emptyset = A_0 \subset A_1 \subset \cdots \subset A_n = U$ is a chain such that

$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{1}_{\mathbf{A_i}}$, and

$\sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0$

key result of Lovász

[Lovász, 1983]

- an input to f is one of the 2^n corners of the n-dimensional unit hypercube
- $\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{1}_{\mathbf{A_i}}$ is an interpolation of the certain vertices of this hypercube
- $f^L(\mathbf{x})$ is the corresponding interpolation of f at sets corresponding to each hypercube vertex
- since f^L is restricted to $[0, 1]$, f^L attains its minimum at the corners
- $f(A)$ is submodular iff its continuous extension $f^L(\mathbf{x})$ is convex

$$\min_{A \subseteq U} f(A) = \min_{\mathbf{x}} \ f^L(\mathbf{x})$$

the Lovász extension

an equivalent definition

- sample a threshold $\theta \in [0, 1]$ uniformly at random
- given sampled θ define the set

$$A_\theta(x) = \{i : x_i > \theta\}$$

- then Lovász extension f^L of f can be defined from

$$f^L(\mathbf{x}) = \mathbf{E}[f(A_\theta(x))]$$

entropy and mutual information

- entropy of a discrete random variable X

$$H(X) = -\sum_{x} \mathbf{Pr}(x) \log \mathbf{Pr}(x)$$

- entropy of X conditioned on Y

$$H(X \mid Y) = -\sum_{x,y} \mathbf{Pr}(x,y) \log \frac{\mathbf{Pr}(x,y)}{\mathbf{Pr}(y)}$$

entropy and mutual information

- mutual information of X and Y: measure of their mutual dependence

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \\ &= H(X,Y) - H(X \mid Y) - H(Y \mid X) \end{aligned}$$

- if X and Y are statistically independent then $I(X;Y) = 0$

entropy and mutual information

- given n random variables $U = \{X_i\}_{i \in [1,n]}$, define

$$f(A) = H(X_A)$$

  to be the joint entropy of the variables indexed by A.

- then f is submodular

entropy and mutual information

- suppose that $A \subseteq B$, $X_e \in U$, then

$$
\begin{aligned}
f(A \cup \{X_e\}) - f(A) &= H(X_A, X_e) - H(X_A) \\
&= H(X_e \mid X_A) \text{ “\textit{information never hurts}"} \\
&\geq H(X_e \mid X_B)
\end{aligned}
$$

- information never hurts: conditioning on data never increases uncertainty
- mutual information is also submodular

$$
I(A) = f(A) + f(U \setminus A) - f(U)
$$

variable selection in classification / regression

- let Y be a random variable we want to predict based on at most n observed measurement variables $X_U = \{X_1, \cdots, X_n\}$
- it might be too costly to use n variables
- goal: choose a subset $A \subseteq U$ variables of size at most k such that predictions based on $\mathbf{Pr}(y \mid x_A)$ retain accuracy

variable selection in classification / regression

- define $f : 2^U \to \mathbb{R}$ as the mutual information function
- $f(A) = I(Y; X_A)$ measures how well variables in A can predict Y
- this means that we want to find A such that $f(A)$ is maximized
- same reasoning directly applicable to sensor coverage and pattern recognition problems

active learning and semi-supervised learning

- given training data $\mathcal{D}_U = \{(x_i, y_i)\}_{i \in U}$ of (x, y) pairs
- often getting y is time-consuming, expensive, and error prone (e.g., Amazon Turk)
- batch active learning: choose a subset $A \subset U$ of size k to acquire the labels $\{y_i\}_{i \in A}$
- adaptive active learning: choose a policy where the decision to select y_i is based on previously chosen labels $\{y_1, \cdots, y_{i-1}\}$, for $i = \{2, \cdots, k\}$

active learning and semi-supervised learning

- goal: choose a subset of k training instances for labeling
- consider the following objective

$$\Psi(A) = \min_{B \subseteq U \setminus A} \frac{\Gamma(B)}{|B|}$$

where

$$\Gamma(B) = I_f(B; U \setminus B) = f(B) + f(U \setminus B) - f(U)$$

is an arbitrary symmetric submodular function

active learning and semi-supervised learning

feature-based learning

- instances represented as feature vectors (what we have been assuming so far)

$$\Gamma(B) = I_f(B; U \setminus B) = f(B) + f(U \setminus B) - f(U)$$

- $\Gamma(B)$: mutual information between B and $U \setminus B$

active learning and semi-supervised learning

graph-based learning learning

- sometimes graph representation is more useful than feature vector representation to exploit relations between instances, e.g., classification of web pages: edge weights can incorporate information about hyperlinks

- feature vector representation can be transformed into graph representation (e.g., by using a Gaussian kernel to compute weights between instances)

active learning and semi-supervised learning

graph-based learning learning

- smoothness assumption: the labels vary smoothly w.r.t. the underlying graph:

$$\sum_{i,j} W_{ij} |y_i - y_j|$$

is small for given weights $\{W_{ij}\}_{(i,j) \in E}$

$$\Gamma(B) = I_f(B; U \setminus B) = f(B) + f(U \setminus B) - f(U)$$

- $\Gamma(B)$: graph cut value between B and $U \setminus B$

active learning and semi-supervised learning

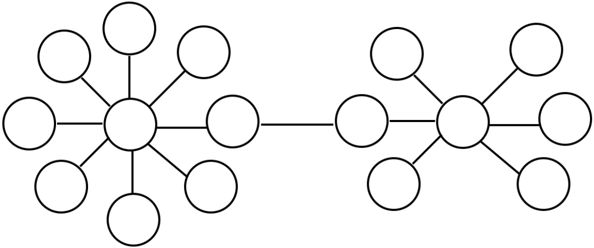
- goal: choose a subset of k training instances for labeling
- consider the following objective [Guillory and Bilmes, 2009]

$$\Psi(A) = \min_{B \subseteq U \setminus A} \frac{\Gamma(B)}{|B|}$$

- small $\Psi(A)$ means an adversary can separate away many (large $|B|$) combinatorially independent (small $\Gamma(B)$) points from A
- small $\Gamma(B)$: low information dependence between B and $U \setminus B$
- this suggests choosing A such that $\Psi(A)$ is maximized

active learning and semi-supervised learning

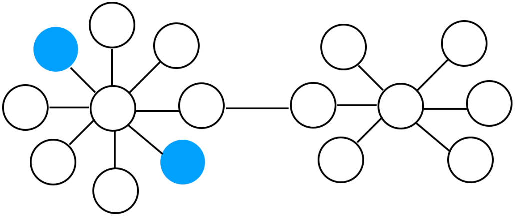
- choose $k = 2$ instances for labeling



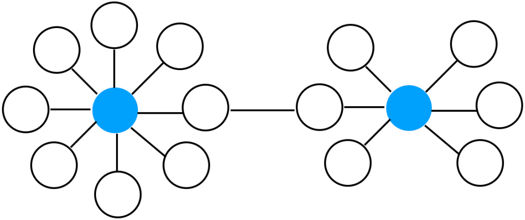
active learning and semi-supervised learning

- which one is better?

  A_1:

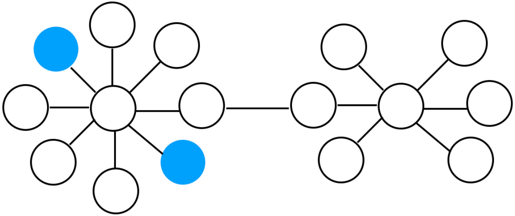


  or A_2:

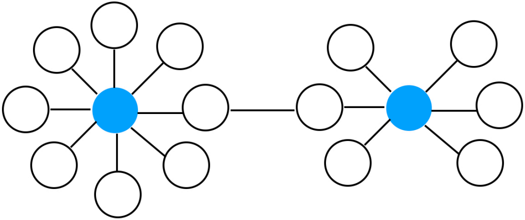


active learning and semi-supervised learning

- $\Psi(A_1) = 1/8$



- $\Psi(A_2) = 1$



active learning and semi-supervised learning

semi-supervised learning

- once we have $\{y_i\}_{i \in A}$, infer the remaining labels $\{y_i\}_{i \in U \setminus A}$
- form a labeling $\mathbf{y}' \in \{0, 1\}^U$ such that $\mathbf{y}'_A = \mathbf{y}_A$, i.e., $\mathbf{y}'$ agrees with the known labels $\mathbf{y}_A$
- $\Gamma(B)$ measures label smoothness, i.e., how much information dependence between labels in B and complement $U \setminus B$

  i.e., graph case: label change should be across small cuts

active learning and semi-supervised learning

semi-supervised learning

- let A^+ denote instances with obtained positive labels
- let $L = U \setminus A$ denote the instances with missing labels
- we want to choose $L^+ \subseteq L$ for assigning positive labels such that $\Gamma(L^+ \cup A^+)$ is minimized

active learning and semi-supervised learning

semi-supervised learning

- this is submodular minimization on the function $g : 2^L \rightarrow \mathbb{R}_+$ where for $L^+ \in U \setminus A$

$$g(L^+) = \Gamma(L^+ \cup A^+)$$

- in graph representation case, this is the standard min-cut approach to semi-supervised learning by [Blum and Chawla, 2001]

learning submodular functions

probably mostly approximately correct (PMAC) learning
[Balcan and Harvey, 2011]

- sample $S = \{(A_1, f(A_1)), \cdots, (A_m, f(A_m))\}$
- learner sees A_i's sampled i.i.d. from distribution $\mathcal{D}$ on 2^U

  and produces a hypothesis h
- goal: with probability at least $1 - \delta$ over the choice of random sample $S \sim \mathcal{D}^m$:

$$\mathbf{Pr}_{A \sim \mathcal{D}}(h(A) \leq f(A) \leq \alpha h(A)) \geq 1 - \epsilon$$

- approximation ratio $\alpha \geq 1$ allows for multiplicative error
- PAC model is special case with $\alpha = 1$

learning submodular functions

probably mostly approximately correct (PMAC) learning

- upper bound: there exists an algorithm for PMAC-learning the class of submodular functions with an approximation factor $\alpha = \mathcal{O}(n^{1/2})$
- lower bound: no algorithm can PMAC-learn the class of submodular functions with an approximation factor $\alpha = \mathcal{O}(n^{1/3})$

references

Balcan, M.-F. and Harvey, N. J. (2011).
Learning submodular functions.
In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 793–802. ACM.

Blum, A. and Chawla, S. (2001).
Learning from labeled and unlabeled data using graph mincuts.
In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 19–26. Morgan Kaufmann Publishers Inc.

Borodin, A., Lee, H. C., and Ye, Y. (2012).
Max-sum diversification, monotone submodular functions and dynamic updates.
In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 155–166. ACM.

references (cont.)

- Feige, U., Mirrokni, V. S., and Vondrak, J. (2011).
  Maximizing non-monotone submodular functions.
  SIAM Journal on Computing, 40(4):1133–1153.
- Fisher, M. L., Nemhauser, G. L., and Wolsey, L. A. (1978).
  An analysis of approximations for maximizing submodular set functions ii.
  In *Polyhedral combinatorics*.
- Guillory, A. and Bilmes, J. A. (2009).
  Label selection on graphs.
  In *Advances in Neural Information Processing Systems*, pages 691–699.

references (cont.)

- Lin, H., Bilmes, J., and Xie, S. (2009).
  Graph-based submodular selection for extractive summarization.
  In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386. IEEE.

- Lovász, L. (1983).
  Submodular functions and convexity.
  In *Mathematical Programming The State of the Art*, pages 235–257. Springer.

- Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978).
  An analysis of approximations for maximizing submodular set functions I.
  Mathematical Programming, 14(1):265–294.