EE596A Submodular Functions **University of Washington**
Spring 2016 **Dept. of Electrical Engineering**

Homework 4. Due **May 25th, 11:59pm** Electronically

Prof: J. Bilmes <bilmes@ee.washington.edu> *Monday, May 2 2016*
TA: K. Wei <kaiwei@uw.edu>

All homework is due electronically via the link https://canvas.uw.edu/courses/1039754/assignments. Note that the due dates and times might be in the evening. Please submit a PDF file. Doing your homework by hand and then converting to a PDF file (by say taking high quality photos using a digital camera and then converting that to a PDF file) is fine, as there are many jpg to pdf converters on the web. Some of the problems below will require that you look at some of the lecture slides at our web page (http://j.ee.washington.edu/~bilmes/classes/ee596b_spring_2016/).

Problem 1. From non-submodular to submodular

Let $h : 2^V \to \mathbb{R}$ be an arbitrary real-valued set function. Consider the functions $g : 2^V \to \mathbb{R}$ and $f : 2^V \to \mathbb{R}$ obtained from h (recursively in the case of g) as follows. For $S \subseteq V$,

$$g(S) = \begin{cases} \min_{A \subset S, B \subset S : A \cup B = S} \Big(\min[h(S), g(A) + g(B) - g(A \cap B)] \Big) & \text{if } |S| \geq 2 \\ h(S) & \text{else .} \end{cases} \tag{1}$$

Note that $g(\emptyset) = h(\emptyset)$ and $g(a) = h(a)$ for all $a \in V$. We also define f as follows:

$$f(S) = \min_{A \supseteq S} g(A). \tag{2}$$

Problem 1(a). submodular Prove that for any h, then g as defined above is submodular.

Problem 1(b). monotone-nondecreasing Prove that for any h, then f as defined above is submodular and monotone non-decreasing.

Problem 1(c). identity Prove that if h is submodular, then $g = h$.

Problem 1(d). monotone identity Prove that if h is submodular and monotone non-decreasing, then $f = h$.

Problem 2. k-medoids clustering Given a ground set of data points $V = \{v_1, \ldots, v_n\}$, let $d_{v_i, v_j} \geq 0$ denote the distance measure between data point v_i and v_j (e.g., squared Euclidean distance between the feature representation of v_i and v_j). Assume that the distance measure is symmetric, i.e., $d_{v_i, v_j} = d_{v_j, v_i}$. The goal of k-medoids clustering is to identify a set $A \subseteq V$ of k medoids such that, by using each point in A as a cluster center, the total within cluster scatter is minimized. Mathematically, the k-medoid clustering problem is formulated as below:

$$\min_{A \subseteq V, |A| = k} c(A), \tag{3}$$

where $c(A) = \sum_{v \in V} \min_{a \in A} d_{a,v}$ is the clustering cost for choosing the set A as the medoids.

Problem 2(a). Monotonicity Determine the monotonicity of the objective function $c(A)$. Note that a set function $c(A)$ is monotonically non-decreasing if $c(A) \leq c(B)$ for any $A \subseteq B$, and $c(A)$ is monotonically non-increasing if the reverse always holds.

Problem 2(b). Supermodularity Prove that the objective function $c(A)$ is supermodular.

Problem 2(c). Connection to submodular maximization Let $d^* = \max_{v \in V, v' \in V} d_{v,v'}$ be the maximum distance between any pair of points in V. We define the similarity measure between any pair of points

v_i and v_j as $s_{v_i,v_j} = d^* - d_{v_i,v_j} \geq 0$. Define a utility function $f(A) = \sum_{v \in V} \max_{a \in A} s_{v,a}$. Prove that the following optimization problem:

$$\max_{A \subseteq V, |A|=k} f(A) \tag{4}$$

is *equivalent* to Problem 3. By equivalence, we mean: any solution A^* for Problem 3 (i.e., $A^* \in \operatorname{argmin}_{A \subseteq V, |A|=k} c(A)$) is also a solution for Problem 4 (i.e., $A^* \in \operatorname{argmax}_{A \subseteq V, |A|=k} f(A)$), and the same holds for the converse.

Problem 3. Matroid constrained submodular maximization In the lecture, we have proved that greedy algorithm solves the matroid constrained modular maximization problem. Given a normalized modular function $m : V \to \mathbb{R}_+$, and a matroid $\mathcal{M}(V, \mathcal{I})$, the greedy heuristic exactly solves the following optimization problem:

$$\max_{A \in \mathcal{I}} m(A). \tag{5}$$

In this problem, we investigate the same greedy algorithm in the case of the objective being polymatroid function $f : 2^V \to \mathbb{R}_+$. The pseudo code for such algorithm is shown below:

Algorithm 1: Greedy algorithm

Input: A polymatroid function f and a matroid $\mathcal{M}(V, \mathcal{I})$.
$A_0 \leftarrow \emptyset$.
$i = 0$.
while A *is not a base of* $\mathcal{M}$ **do**
  $i \leftarrow i + 1$.
  $a_i \in \operatorname{argmax}_{a \in V \setminus A_{i-1} : A_{i-1} + a \in \mathcal{I}} f(a | A_{i-1})$.
  $A_i \leftarrow A_{i-1} + a_i$.
Output $\hat{A} \leftarrow A_i$.

We will prove that such algorithm always yields a solution with an approximation factor $1/2$, namely, the following holds:

$$f(\hat{A}) \geq \frac{1}{2} \max_{A \in \mathcal{I}} f(A). \tag{6}$$

Problem 3(a). Please first show that the following statement is true: Given any base $B = \{b_1, \ldots, b_k\}$ of the matroid $\mathcal{M}$ (assuming the rank of the matroid $\mathcal{M}$ is k), the elements in the base B can be ordered such that $f(b_i | A_{i-1}) \leq f(a_i | A_{i-1})$ for all $i = 1, \ldots, k$.

Problem 3(b). Define A^* to be the optimal solution, i.e., $A^* \in \operatorname{argmax}_{A \in \mathcal{I}} f(A)$. Show that the following always holds:

$$f(A^*) \leq f(A_k) + \sum_{i : a_i \in A^* \cap A_k} f(a_i | A_{i-1}) + \sum_{a \in A^* \setminus A_k} f(a | A_k). \tag{7}$$

Problem 3(c). Prove the $1/2$ approximation guarantee, i.e.,

$$f(\hat{A}) \geq \frac{1}{2} \max_{A \in \mathcal{I}} f(A) \tag{8}$$

Problem 4. Visualize the facility location function

You are provided with four sets of 2-dimensional data points $V = \{(x_i, y_i)\}_i$ (see the files "data_set{1,2,3,4}.txt" for the data). Each data file is formatted such that each line represents a data point. For each line of the file, the first entry is the value for the x-axis, and the second entry defines the y-axis value.

In this exercise, you will understand how the facility location function can be used to choose a set of k representative data points in the ground set V. To finish this problem, you will write the code in your favorite programming language to implement the greedy algorithm on the facility location function. The pseudo code for the greedy algorithm is as follows:

Algorithm 2: Greedy algorithm

Input: f, V, and k.
Initialization: $A \leftarrow \emptyset$.
while $|A| < k$ **do**
  $a^* \in \operatorname{argmax}_{a \in V \setminus A} f(a|A)$
  $A \leftarrow A \cup a^*$
Output: A.

Problem 4(a). Plot the 2-D data Your task here is to plot out the data points in V on a 2 dimensional plot for each data set separately. In your plot, please make sure to label the x-axis and y-axis clearly.

Problem 4(b). Gaussian kernel as similarity

Note that the facility location used as the objective in this problem is defined as below:

$$f_{\text{fac}}(A) = \sum_{i \in V} \max_{j \in A} w_{i,j}, \tag{9}$$

where $w_{i,j}$ is the similarity measure between data points i and j. It remains to define the similarity between data points from their distance measure. In this exercise we use the simple Euclidean distance as the distance measure, i.e., we define $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Given σ as the width parameter for the Gaussian kernel, we define the Gaussian kernel between any pair of points i and j as $w^{\sigma}_{i,j} = \exp(-\frac{d^2_{i,j}}{2\sigma^2})$.

For each data set, run the greedy algorithm with a size constraint $k = 10$ on the facility location function whose similarity measure is computed as the Gaussian kernel with the choice of the width parameter $\sigma^2 = 1$. Mark each chosen data point in the 2-D plot with a circle and report the facility location function value of the output of the greedy algorithm.

Repeat the previous procedure for other choices of the width parameter: $\sigma^2 = 0.1, 10$, and show the plot for each case of σ.

Problem 4(c). Linear kernel as similarity In this problem, we try out deriving the similarity measure from the Euclidean distance in a different way. In this case, we define the similarity measure as $w_{i,j} = -d_{i,j}$. Run the greedy algorithm with $k = 10$ on this variant of the facility location function. Make a plot of all data points in the 2-D space with each of the chosen items marked in a circle.

Problem 5. Stochastic Variants of Greedy algorithms

Recall from the class, we have talked about a simple greedy algorithm for solving the following problem:

$$\max_{|A| \leq k} f(A), \tag{10}$$

where f is monotone submodular and k is the size constraint. The algorithm is described below:

Algorithm 3: Greedy algorithm

Input: f and k.
Initialization: $A_0 \leftarrow \emptyset$ and $i = 0$
while $i < k$ **do**
  $a_{i+1} \in \text{argmax}_{a \in V \setminus A} f(a|A)$
  $A_{i+1} \leftarrow A_i \cup a_{i+1}$
  $i \leftarrow i + 1$
Output A_k.

In this problem, we consider several stochastic variants of the greedy algorithm, and analyze their optimality guarantees.

Problem 5(a). Given an instance of running a randomized algorithm ALG which produces a chain of solutions $A_1 \subset A_2 \subset, \ldots, \subset A_k$. Denote $A_k = \{a_1, \ldots, a_k\}$ with a_i being the item added in round i. Consider any round i, denote the expected function gain conditioned on the solution A_{i-1} as $\mathbb{E}[f(a_i|A_{i-1})]$. Suppose that the following holds

$$\mathbb{E}[f(a_i|A_{i-1})] \geq \frac{f(OPT) - f(A_{i-1})}{k} \tag{11}$$

for all i, where $OPT \in \text{argmax}_{|A| \leq k} f(A)$.

Prove the following:

$$\mathbb{E}[f(A_k)] \geq (1 - 1/e) f(OPT). \tag{12}$$

Problem 5(b). Armed with the above result, we are now ready to analyze the following three stochastic variants of the greedy algorithm. Please show, for each algorithm below, whether the approximation factor of $(1 - 1/e)$ on expectation can be achieved.

Algorithm 4: Stochastic Greedy 1

Input: f and k.
Initialization: $A_0 \leftarrow \emptyset$ and $i = 0$
while $i < k$ **do**
  $B^* \in \text{argmax}_{B \subseteq V \setminus A_i, |B| = k} \sum_{b \in B} f(b|A_i)$
  Uniformly at random sample an item a_{i+1} from B^*
  $A_{i+1} \leftarrow A_i \cup a_{i+1}$
  $i \leftarrow i + 1$
Output A_k.

Algorithm 5: Stochastic Greedy 2

Input: f and k.
Initialization: $A_0 \leftarrow \emptyset$ and $i = 0$
while $i < k$ **do**
  $B^* \in \text{argmax}_{B \subseteq V \setminus A_i, |B| = k} f(B|A_i)$
  Uniformly at random sample an item a_{i+1} from B^*
  $A_{i+1} \leftarrow A_i \cup a_{i+1}$
  $i \leftarrow i + 1$
Output A_k.

Problem 5(c). Lastly, we consider another variant of the stochastic greedy algorithm. Note that the line $B^* \in \text{argmax}_{B \subseteq V \setminus A_i, |B| = k} f(B|A_i)$ in Stochastic Greedy 2 is not feasible to solve exactly. One

Algorithm 6: Stochastic Greedy 3

Input: f and k.
Initialization: $A_0 \leftarrow \emptyset$ and $i = 0$
while $i < k$ **do**
  $B^* \in \operatorname{argmax}_{B \subseteq V \setminus A_i, |B|=k} \min_{b \in B} f(b|A_i)$
  Uniformly at random sample an item a_{i+1} from B^*
  $A_{i+1} \leftarrow A_i \cup a_{i+1}$
  $i \leftarrow i + 1$
Output A_k.

may wish to approximately solve this line with a greedy algorithm leading to Stochastic Greedy 4 described below. Show that Stochastic Greedy 4 always attains a guarantee of $(1 - e^{-(1-1/e)})$ on expectation.

Algorithm 7: Stochastic Greedy 4

Input: f and k.
Initialization: $A_0 \leftarrow \emptyset$ and $i = 0$
while $i < k$ **do**
  $\hat{B}$ is obtained by running the greedy algorithm for solving $\operatorname{argmax}_{B \subseteq V \setminus A_i, |B|=k} f(B|A_i)$
  Uniformly at random sample an item a_{i+1} from $\hat{B}$
  $A_{i+1} \leftarrow A_i \cup a_{i+1}$
  $i \leftarrow i + 1$
Output A_k.