Sequential Facility Location: Approximate Submodularity and Greedy Algorithm

Ehsan Elhamifar [1]

Abstract

We develop and analyze a novel utility function and a fast optimization algorithm for subset selection in sequential data that incorporates the dynamic model of data. We propose a cardinality-constrained sequential facility location function that finds a fixed number of representatives, where the sequence of representatives is compatible with the dynamic model and well encodes the data. As maximizing this new objective function is NP-hard, we develop a fast greedy algorithm based on submodular maximization. Unlike the conventional facility location, the computation of the marginal gain in our case cannot be done by operations on each item independently. We exploit the sequential structure of the problem and develop an efficient dynamic programming-based algorithm that computes the marginal gain exactly. We investigate conditions on the dynamic model, under which our utility function is (ε-approximately) submodualr, hence, the greedy algorithm comes with performance guarantees. By experiments on synthetic data and the problem of procedure learning from instructional videos, we show that our framework significantly improves the computational time, achieves better objective function values and obtains more coherent summaries.

1. Introduction

Subset selection, which is to find a small subset of most informative items from a large ground set, is a fundamental task in machine learning with numerous applications, including, image, video, speech and document summarization (Gong et al., 2014; Simon et al., 2007; Elhamifar & De-Paolis-Kaluza, 2017a; Lin & Bilmes, 2012; Kulesza & Taskar, 2012; Frey & Dueck, 2007), data clustering (Kim et al., 2011; Shah & Ghahramani, 2013; Elhamifar et al., 2016), feature and model selection (Guyon & Elisseeff, 2003; Misra et al., 2014; Elhamifar et al., 2014), sensor placement (Krause et al., 2008; Joshi & Boyd, 2009), social network marketing (Hartline et al., 2008) and product recommendation (McSherry, 2002). From an optimization perspective, subset selection consists of two parts. The first component is the utility function, which characterizes the informativeness of selected items. Different criteria have been proposed in the literature, such as maximum cut (Hadlock, 1975; Motwani & Raghavan, 1995), maximum marginal relevance (Carbonell & Goldstein, 1998), capacitated and uncapacitated facility location (Mirchandani & Francis, 1990; Nemhauser et al., 1978), linear coding (Elhamifar et al., 2012a; Esser et al., 2012) and maximum volume parallelotope (Kulesza & Taskar, 2012; Borodin & Olshanski, 2000). The second component is the algorithm to optimize the utility function. In fact, optimizing almost all subset selection criteria is, in general, non-convex and NP-hard (Motwani & Raghavan, 1995; Feige, 1998; Gonzalez, 1985; Civril & Magdon-Ismail, 2009), which has motivated approximate methods, such as greedy algorithms for maximizing submodular functions (Nemhauser et al., 1978; Krause & Golovin, 2014), e.g., graph-cuts and facility location, sampling from Determinantal Point Process (DPP) (Kulesza & Taskar, 2012; Borodin & Olshanski, 2000) for approximately finding the maximum volume subset of points, as well as convex relaxation for subset selection (Elhamifar et al., 2016; Awasthi et al., 2015; Nellore & Ward, 2015; Elhamifar et al., 2012b) .

Sequential Subset Selection: Sequential data, including time-series and ordered data such as video, audio, sensor measurements, text and gene expressions, constitute a large part of today's real-world data. There are two characteristics of sequential data that must be taken into account when developing subset selection techniques. The first is the structured dependencies among data, imposed by underlying dynamic models. For instance, segments/sentences in a video/text are connected in logical way, hence, cannot be treated independently, which would result in losing the semantic content of the video/document. The second characteristic is the large scale of sequential data that needs

[1]Assistant Professor, Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA. Correspondence to: Ehsan Elhamifar <e.elhamifar@northeastern.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

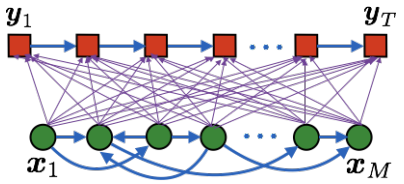


Figure 1: Given a source set of items $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ with a dynamic transition model and a target set of sequential items $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$, we propose the cardinality-constrained sequential facility location function and a fast greedy algorithm to find a sequence of representatives, of fixed cardinality, from the source that well encodes the target data and follows the dynamic model.

to be dealt with efficiently, as we are constantly capturing data, such as video, audio and text.These factors motivate the development of new utility functions and optimization algorithms that take into account structured dependencies of and scale to large sequential data.

The majority of existing subset selection methods, however, treat items independent from each other, ignoring dependencies of sequential items. The work in (Affandi et al., 2012; Gong et al., 2014) have studied interesting extensions of DPP-based subset selection, by finding representatives in a sequential fashion such that newly selected representatives are diverse with respect to the previously selected ones. However, sequential diversity by itself is generally insufficient, e.g., it results in selecting irrelevant but diverse scenes/sentences in video/document summarization. (Tschiatschek et al., 2017) has extended submodular functions to capture ordered preferences among items, where ordered preferences are represented by a directed acyclic graph over items, and has developed a greedy algorithm to pick edges instead of items. However, it cannot deal with arbitrary graphs, such as the ones with cycles. The recent work in (Elhamifar & De-Paolis-Kaluza, 2017b) has introduced a method for sequential subset selection that incorporates dynamics of data. However, the proposed message passing algorithm has $O(N^4)$ computational cost in the size of data, N, making it impractical to apply to large datasets. Moreover, it has no explicit control over the exact number of representatives, which is desired in many real problems.

Paper Contributions: We develop a new utility function and a fast optimization algorithm for subset selection in sequential data. We propose a cardinality-constrained sequential facility location function that incorporates dynamics of data into subset selection, finding a fixed number of representatives, where the sequence of representatives well encodes the data and is compatible with the dynamic model, see Figure 1. As maximizing this new objective function is, in general, NP-hard, we develop a fast greedy algorithm based on submodular maximization. Unlike conventional facility location, due to the coupling of representatives via the dynamic model, the computation of the marginal gain cannot be done by operations on each item independently. Hence, we exploit the sequential structure of the problem and develop a dynamic programming-based algorithm to compute the marginal gain exactly. We investigate conditions on the dynamics of data, under which our utility function is (ε-approximately) submodular, hence, the greedy algorithm comes with performance guarantees. For a first order Markov model, our theoretical conditions depend on the variations of incoming transitions to each item, where for a uniform transition the problem would become submodular. By experiments on synthetic and on the problem of procedure learning from instructional videos, we show that our framework not only significantly improves the computational time, but also achieves better objective function values and more coherent summaries than existing methods.

2. Background Review

In this section, we review the sequential subset selection algorithm in (Elhamifar & De-Paolis-Kaluza, 2017b) and approximate submodular functions. To handle large sequential data, we build on these and introduce a new utility function, a fast algorithm and study its theoretical guarantees.

2.1. Sequential Subset Selection

Consider a source set of items $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ with a first-order Markov model transition dynamics, $\pi(\boldsymbol{x}_{i'}|\boldsymbol{x}_i)$, and a target set of sequential items $\mathcal{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$. The goal of sequential subset selection is to find a representative subset of $\mathcal{X}$ that well encodes $\mathcal{Y}$, while the sequence of representatives follow the dynamic model on $\mathcal{X}$. The recent work in (Elhamifar & De-Paolis-Kaluza, 2017b) has proposed a formulation for this problem based on a generalization of the *uncapacitated* facility location function.

Let $d_{i,t}$ denote the dissimilarity of $\boldsymbol{x}_i$ to $\boldsymbol{y}_t$. A lower $d_{i,t}$ means that $\boldsymbol{x}_i$ better encodes $\boldsymbol{y}_t$. Define binary assignment variables $\{z_{i,t}\}_{i=1,\ldots,M}^{t=1,\ldots,T}$, where $z_{i,t} \in \{0,1\}$ indicates if $\boldsymbol{x}_i$ is a representative of $\boldsymbol{y}_t$. Assuming that each item $\boldsymbol{y}_t$ is associated with only one representative, we have $\sum_{i=1}^{M} z_{i,t} = 1$. To recover the assignment variables and representatives, (Elhamifar & De-Paolis-Kaluza, 2017b) proposes to solve

$$\begin{aligned}
&\min_{\{z_{i,t}\}} \sum_{t=1}^{T}\sum_{i=1}^{M} d_{i,t} z_{i,t} + \lambda \sum_{i=1}^{M} \| \begin{bmatrix} z_{i,1} & \cdots & z_{i,T} \end{bmatrix} \|_{\infty} \\
&- \beta \Big(\sum_{i=1}^{M} \log \pi_0(\boldsymbol{x}_i) z_{i,1} + \sum_{t=2}^{T} \sum_{i,i'=1}^{M} \log \pi(\boldsymbol{x}_{i'}|\boldsymbol{x}_i) z_{i,t-1} z_{i',t} \Big) \\
&\text{s.t.} \quad z_{i,t} \in \{0,1\}, \quad \sum_{i=1}^{M} z_{i,t} = 1, \ \forall\, i, t,
\end{aligned} \tag{1}$$

where the first term measures the encoding cost of $\mathcal{Y}$ via representatives, since we pay the penalty of $d_{i,t}$ for $\boldsymbol{x}_i$ being

the representative of $\boldsymbol{y}_t$. The second term in the objective function counts the number of representatives, since $\| \begin{bmatrix} z_{i,1} & \cdots & z_{i,T} \end{bmatrix} \|_\infty$ would be 1 if $\boldsymbol{x}_i$ becomes a representative of some items in $\mathcal{Y}$ and would be 0 otherwise. Finally, the last two terms of the objective function measure the compatibility of the sequence of representatives. The term $\sum_{i=1}^{M} \log \pi_0(\boldsymbol{x}_i) z_{i,1}$ promotes to select an $\boldsymbol{x}_i$ as the first representative, i.e., representative of $\boldsymbol{y}_1$, that has a large initial probability π_0. The term $\sum_{i,i'=1}^{M} \log \pi(\boldsymbol{x}_{i'}|\boldsymbol{x}_i) z_{i,t-1} z_{i',t}$ promotes to select i and i' as representatives of consecutive items of $\mathcal{Y}$, when there is a high transition probability between them. As a result, incorporating the dynamic compatibility term promotes to select a sequence of representatives that follow the dynamic model. The regularization parameter $\lambda > 0$ controls the effect of the cardinality term, where a close to zero λ results in many representatives, while the regularization parameter $\beta > 0$ controls the effect of the dynamic compatibility term, where a close to zero β results in discounting the effect of the dynamics on $\mathcal{X}$.

Since (1) is non-convex, due to the dynamic cost and the binary constraints on assignment variables, (Elhamifar & De-Paolis-Kaluza, 2017b) proposes a max-sum message passing algorithm to approximately solve the problem. However, a major drawback of this approach is that the message passing algorithm (on the corresponding loopy graph) is computationally complex, requiring $O(M^2T^2 + M^3T)$ computations per iteration of the message passing, hence, does not scale to large sequential data. Moreover, it is not clear how to choose λ in order to find a certain number of representatives, as is often desired in real applications. In the next section, we propose a cardinality-constrained sequential facility location function that finds a fixed number of representatives and study a fast greedy algorithm, which significantly reduces the computational cost to $O(MT)$.

2.2. Greedy Submodular Maximization

Consider a set function f defined on a ground set $\mathcal{V}$, i.e., a function that assigns a value $f(\mathcal{S})$ to each subset $\mathcal{S} \subseteq \mathcal{V}$. The function is submodular if it satisfies the diminishing return property (Krause & Golovin, 2014).

Definition 1 *A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular if for every $\Lambda \subseteq \Gamma \subseteq \mathcal{V}$ and every $i \in \mathcal{V} \backslash \Gamma$, we have $f(\Lambda \cup \{i\}) - f(\Lambda) \geq f(\Gamma \cup \{i\}) - f(\Gamma)$.*

Roughly speaking, the diminishing return property states that the added value of including an element to a set decreases as the size of the set increases. To maximize a set function, we often use the marginal gain, defined as follows.

Definition 2 *The marginal gain of a set function $f : 2^{\mathcal{V}} \to \mathbb{R}$, at Λ with respect to $i \in \mathcal{V}$ is defined as*

$$\delta_f(i|\Lambda) \triangleq f(\Lambda \cup \{i\}) - f(\Lambda). \tag{2}$$

Given a submodular set function f, maximization of f over all subsets of size at most k of the ground set, i.e.,

$$\max_{\mathcal{S}:|\mathcal{S}| \leq k} f(\mathcal{S}), \tag{3}$$

is an NP-hard problem (Feige, 1998), (Gonzalez, 1985). An approximate method to solve this problem is a greedy algorithm (Nemhauser et al., 1978) that starts by initializing an active set Λ to be empty and, over k iterations, incrementally adds to Λ the element that maximizes the marginal gain. Algorithm 1 shows the steps of the greedy method. For monotone submodular functions, i.e., functions for which $f(\Lambda) \leq f(\Gamma)$ for any $\Lambda \subseteq \Gamma$, the greedy algorithm is guaranteed to obtain a solution that is within $(1 - 1/e)$ of the global maximum of (3) (Nemhauser et al., 1978), (Calinescu et al., 2011), (Feldman et al., 2011). The recent result in (Bian et al., 2017) has shown that even for non-submodular functions, the greedy algorithm has worst case guarantees, which depend on the generalized curvature and the submodularity ratio of the function.

Algorithm 1 : Greedy Maximization

Input: Set function $f : 2^{\mathcal{V}} \to \mathbb{R}$; Budget k.

1: Initialize: $\Lambda = \varnothing$;
2: **for** $j = 1, \ldots, k$ **do**
3: **for** $i \in \mathcal{V} \backslash \Lambda$ **do**
4: Compute $\delta_f(i|\Lambda) \triangleq f(\Lambda \cup \{i\}) - f(\Lambda)$;
5: **end for**
6: Compute $i^* = \text{argmax}_{i \in \mathcal{V} \backslash \Lambda} \delta_f(i|\Lambda)$;
7: Update $\Lambda \leftarrow \Lambda \cup \{i^*\}$;
8: **end for**

Output: Optimal set Λ of size k.

A more general class of set functions is ε-approximately submodular function that also has approximation guarantees via the standard greedy algorithm (Horel & Singer, 2016).

Definition 3 *A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is ε-approximately submodular, if there exists a submodular function $g : 2^{\mathcal{V}} \to \mathbb{R}$, such that for every $\Lambda \subseteq \mathcal{V}$,*

$$(1 - \varepsilon) g(\Lambda) \leq f(\Lambda) \leq (1 + \varepsilon) g(\Lambda). \tag{4}$$

Indeed, solving (3) for an ε-approximately submodular function using the standard greedy algorithm has $(1 - 1/e - O(\delta))$ approximation guarantees, where $\delta = \varepsilon k$ (Horel & Singer, 2016).

3. Cardinality Constrained Sequential Facility Location

In this section, we propose a cardinality-constrained maximization for sequential subset selection, develop a fast greedy algorithm that maximizes the marginal gain via dynamic programming, and study conditions under which our utility function is (approximately) submodular.

3.1. Problem Formulation

Given a source set $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ with a first-order Markov model dynamics, (π_0, π), and a sequential target set $\mathcal{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_T)$, our goal is to find a representative subset of $\mathcal{X}$ of size at most k that well encodes $\mathcal{Y}$, where the sequence of representatives follows the dynamic model. We consider a clustering-based subset selection scheme, where each $\boldsymbol{y}_t$ is assigned to one representative from $\mathcal{X}$. Here, $\pi_0(i)$ indicates the probability of selecting $\boldsymbol{x}_i$ as the representative of $\boldsymbol{y}_1$ and $\pi(i'|i)$ denotes the probability of selecting $\boldsymbol{x}_{i'}$ as the representative of $\boldsymbol{y}_t$ given that $\boldsymbol{x}_i$ has been selected as the representative of $\boldsymbol{y}_{t-1}$.

Let $\mathcal{S} \subseteq \{1, \dots, M\}$ denote the set of indices of representatives. Let $r_t \in \mathcal{S}$ be the unknown index of the representative of $\boldsymbol{y}_t$ for $t \in \{1, \dots, T\}$ and let $\boldsymbol{r} \triangleq (r_1, r_2, \dots, r_T)$ denote the sequence of representatives for $\mathcal{Y}$. To recover $\mathcal{S}$ and $\boldsymbol{r}$, we propose the maximization

$$\max_{\mathcal{S}:\, |\mathcal{S}| \le k} \; \max_{\boldsymbol{r} \in \mathcal{S}^T} \; \Phi_{\text{enc}}(\boldsymbol{r}) \times \Phi_{\text{dyn}}^{\beta}(\boldsymbol{r}), \tag{5}$$

over all subsets $\mathcal{S} \subseteq \{1, \dots, M\}$ of size k and over all possible assignments $\boldsymbol{r} \subseteq \mathcal{S}^T$, where $\Phi_{\text{enc}}(\cdot)$ denotes an encoding potential that favors selecting a representative set from $\mathcal{X}$ that well encodes $\mathcal{Y}$ and $\Phi_{\text{dyn}}(\cdot)$ denotes a dynamic potential that favors selecting an ordered set of representatives that follows the dynamic model (π_0, π). The regularization parameter $\beta \ge 0$ sets the effect of the dynamic potential, where a close to zero β discounts the dynamics.

Since the encoding of each item of $\mathcal{Y}$ depends on its own representative, we consider the factorization

$$\Phi_{\text{enc}}(\boldsymbol{r}) = \prod_{t=1}^{T} \exp(s_{r_t,t}), \tag{6}$$

where $s_{i,t} \ge 0$ denotes the similarity of $\boldsymbol{x}_i$ to $\boldsymbol{y}_t$, which is larger when $\boldsymbol{x}_i$ better represents $\boldsymbol{y}_t$. For the dynamic potential, we consider

$$\Phi_{\text{dyn}}(\boldsymbol{r}) = \pi_0(r_1) \times \prod_{t=2}^{T} \pi(r_t|r_{t-1}), \tag{7}$$

where the selection of the representative of $\boldsymbol{y}_t$ depends on the representative of $\boldsymbol{y}_{t-1}$[1]. As a result, maximizing the dynamic potential promotes to select a sequence of representatives that follow the dynamic model on the source set. Here, we assume that the dynamic model is given. In the real experiments, we learn (π_0, π) by fitting a hidden Markov model to data. To simplify the notation, we define

$$q_i^0 \triangleq \log \pi_0(i), \quad q_{i,i'} \triangleq \log \pi(i'|i). \tag{8}$$

[1]We can generalize the method to higher order Markov models. We focus on first-order models as they work well in practice.

Instead of maximizing the utility function in (5), we equivalently maximize its logarithm. Using the definition of the encoding and dynamic potentials in (6) and (7), we get

$$\max_{\mathcal{S}:\, |\mathcal{S}| \le k} \; \max_{\boldsymbol{r} \in \mathcal{S}^T} \; \sum_{t=1}^{T} s_{r_t,t} + \beta\big(q_{r_1}^0 + \sum_{t=2}^{T} q_{r_{t-1},r_t}\big). \tag{9}$$

Without loss of generality, we can shift the values of log initial and log transition probabilities to become positive, by adding a constant to each vector/matrix. This would make the objective function always non-negative, while does not change the optimal solution, as it is equivalent to adding a constant to the entire objective function.

Remark 1 *Instead of using (7), we can define*

$$\Phi_{dyn}(\boldsymbol{r}) = e^{q_{r_1}^0} \times \prod_{t=2}^{T} e^{q_{r_{t-1},r_t}}, \tag{10}$$

where the vector q^0 and the matrix q correspond to the preference or score of initial and transition in representative assignments, respectively. q^0 and q could be defined by the user or be estimated from data. For example, we can set $q_{r_1}^0 = \pi_0(r_1)$ and $q_{r_{t-1},r_t} = \pi(r_t|r_{t-1})$.

Remark 2 *For $\beta = 0$, the utility function in (9) reduces to the cardinality-constrained facility location, which is submodular and for which the standard greedy algorithm has $(1 - 1/e)$ worst case performance guarantees. In this case, if we know the optimal representative set $\mathcal{S}$, computing r_t can be done independently for each t by assigning the closest element from $\mathcal{S}$ to $\boldsymbol{y}_t$, i.e., $r_t = \text{argmax}_{i \in \mathcal{S}}\, s_{i,t}$. On the other hand, for $\beta > 0$, it is not clear if the utility function in (9) is submodular. Moreover, computing the assignment variables cannot be done independently over t, due to the coupling of the sequence of representatives via π (or q).*

Next, we use a greedy linear time algorithm based on dynamic programming in each iteration and investigate conditions under which our utility function is (approximately) submodular, hence, has approximation guarantees.

3.2. Greedy Sequential Facility Location

To solve our proposed optimization in (9), we first rewrite it in the form of the maximization of a set function as in (3), by defining f over the ground set $\mathcal{V} \triangleq \{1, \dots, M\}$ as

$$f(\mathcal{S}) \triangleq \max_{\boldsymbol{r} \in \mathcal{S}^T} \; \sum_{t=2}^{T} w_{r_{t-1},r_t}, \tag{11}$$

where, for compactness of notation, for every t in $\{2, \dots, T\}$, we have defined

$$w_{r_{t-1},r_t} \triangleq \begin{cases} s_{r_{t-1},t-1} + \beta(q_{r_{t-1}}^0 + q_{r_{t-1},r_t}), & \text{if } t = 2, \\ s_{r_{t-1},t-1} + \beta q_{r_{t-1},r_t}, & \text{if } 2 < t < T, \\ s_{r_{t-1},t-1} + s_{r_t,t} + \beta q_{r_{t-1},r_t}, & \text{if } t = T. \end{cases} \tag{12}$$

Notice that $f(\varnothing) = 0$ and it is easy to verify that f is monotone, i.e., $f(\Lambda) \leq f(\Gamma)$ for any $\Lambda \subseteq \Gamma$. To solve (9) via the greedy method in Algorithm 1, we need to compute the marginal gain, $\delta_f(i|\Lambda)$, for f defined in (11), which in turn requires evaluations of f on Λ and $\Lambda \cup \{i\}$, see (2).

Evaluation of $f(\Lambda)$, in principle, requires a search over an exponentially large parameter space, $\boldsymbol{r} \subseteq \Lambda^T$, and, unlike the case with $\beta = 0$, cannot be done independently over each time instant. However, we use the sequential structure of the problem to efficiently evaluate f. More specifically, we use the fact that computing the right hand side of (11) involves maximizations over sum of pairs of variables. Hence, we can distribute the maximization over the summation, i.e.,

$$f(\Lambda) = \max_{r_T \in \Lambda} (w_{r_{T-1},r_T} + \cdots + \max_{r_2 \in \Lambda} (w_{r_2,r_3} + \max_{r_1 \in \Lambda} w_{r_1,r_2})). \tag{13}$$

Thus, we can exactly compute $f(\Lambda)$ and $f(\Lambda \cup \{i\})$, hence, the marginal gain, in each iteration of the greedy algorithm using dynamic programming (Bellman, 2003).

Computational Complexity. Notice that at each iteration of the greedy, computing $f(\Lambda \cup \{i\})$ via (13) requires performing T maximizations, where each maximization requires computing a table with $(|\Lambda| + 1)^2$ pairs. Given that $|\Lambda|^2$ pairs have already been computed in the previous iteration of the greedy, we need to compute $O(|\Lambda|)$ new values, hence, $O(kT)$ cost for computing (13), given that $|\Lambda| \leq k$. At each iteration, we need to compute the marginal gain for each i in $\mathcal{V} \backslash \Lambda$, hence, $O(kMT)$ cost per greedy iteration. Finally, to find k representatives, we need to run the greedy algorithm for k iterations, hence, a total cost of $O(k^2MT)$. Thus, our algorithm runs linearly in the size of the source and target sets as opposed to the $O(M^2T^2 + M^3T)$ cost of the message passing in (Elhamifar & De-Paolis-Kaluza, 2017b). Moreover, given a budget k, the greedy algorithm returns at most k representatives, as opposed to the message-passing framework that needs to be run for many λ's in (1) and select the solution that satisfies the budget.

3.3. Theoretical Analysis

Our proposed formulation in (9), involves maximization over a set function $f(\mathcal{S})$, defined as

$$f(\mathcal{S}) \triangleq \max_{\boldsymbol{r} \in \mathcal{S}^T} \sum_{t=1}^{T} s_{r_t,t} + \beta \big(q^0_{r_1} + \sum_{t=2}^{T} q_{r_{t-1},r_t}\big). \tag{14}$$

It is important to note that the sequential facility location in (14) is not necessarily submodular, unlike the standard facility location. To see this, consider the following example.

Example 1 *Consider a source set $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ and a sequential target $\mathcal{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$, with the following similarities and dynamic transition model, for $\nu \leq 0.5$,*

$$s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \pi_0 = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \ \pi = \begin{bmatrix} \nu & 1-\nu \\ 1-\nu & \nu \end{bmatrix}.$$

Consider the shifted log-probabilities,

$$q^0 = \begin{bmatrix} 0.1 & 0.1 \end{bmatrix}, \ q = \begin{bmatrix} 0.1 & \log(\frac{1-\nu}{\nu}) + 0.1 \\ \log(\frac{1-\nu}{\nu}) + 0.1 & 0.1 \end{bmatrix}.$$

Let $\Lambda = \varnothing$ and $\Gamma = \{2\}$. We have

$$\begin{aligned} f(\Lambda) &= 0, \\ f(\Lambda \cup \{1\}) &= s_{1,1} + s_{1,2} + \beta q^0_1 + \beta q_{1,1} = 1 + 0.2\beta, \\ f(\Gamma) &= s_{2,1} + s_{2,2} + \beta q^0_2 + \beta q_{2,2} = 1 + 0.2\beta, \\ f(\Gamma \cup \{1\}) &= s_{1,1} + s_{2,2} + \beta q^0_1 + \beta q_{1,2} \\ &= 2 + \beta \log(\frac{1-\nu}{\nu}) + 0.2\beta. \end{aligned} \tag{15}$$

When $0.2 < \log(\frac{1-\nu}{\nu})$, we have $\delta_f(\{1\}|\Lambda) = 1 + 0.2\beta < \delta_f(\{1\}|\Gamma) = 1 + \beta \log(\frac{1-\nu}{\nu})$. Thus, $f(\cdot)$ does not satisfy the diminishing return property and is not submodular.

We investigate conditions on the transition model, π, under which our utility function is (ε-approximately) submodular, hence, the greedy algorithm has approximation guarantees.

Theorem 1 *Consider the optimization in (9). Assume there exists $\varepsilon \in [0, 1)$ such that each q_{r_{t-1},r_t} can be written as $q_{r_{t-1},r_t} = \bar{q}_{r_t} \psi_{r_{t-1},r_t}$, for some $\bar{q}_{r_t}$ where $\psi_{r_{t-1},r_t} \in [1-\varepsilon, 1+\varepsilon]$. Then, for every $\beta \geq 0$, the proposed utility function in (14) is ε-approximately submodular. Moreover, the greedy algorithm has $(1 - 1/e - O(\delta))$ approximation guarantees, where $\delta = \varepsilon k$.*

Proof. To prove the result, we show that there exists a submodular function $g(\mathcal{S})$, such that $f(\mathcal{S})$ satisfies the definition of ε-approximate submodularity via $g(\mathcal{S})$, as in (3). Using the assumption of the theorem, we rewrite (14) as

$$f(\mathcal{S}) = \max_{\boldsymbol{r} \in \mathcal{S}^T} (s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta \bar{q}_{r_t} \psi_{r_{t-1},r_t})). \tag{16}$$

Since from the assumption $\psi_{r_{t-1},r_t} \in [1-\varepsilon, 1+\varepsilon]$, we can bound $f(\mathcal{S})$ from above by

$$\begin{aligned} f(\mathcal{S}) &\leq \max_{\boldsymbol{r} \in \mathcal{S}^T} (s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta(1+\varepsilon)\bar{q}_{r_t})) \\ &\leq (1+\varepsilon) \max_{\boldsymbol{r} \in \mathcal{S}^T} (s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta \bar{q}_{r_t})), \end{aligned} \tag{17}$$

where we used the fact that similarities, log initial and transition probabilities are non-negative. Similarly, we can bound $f(\mathcal{S})$ from below as

$$\begin{aligned} f(\mathcal{S}) &\geq \max_{\boldsymbol{r} \in \mathcal{S}^T} (s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta(1-\varepsilon)\bar{q}_{r_t})) \\ &\geq (1-\varepsilon) \max_{\boldsymbol{r} \in \mathcal{S}^T} (s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta \bar{q}_{r_t})). \end{aligned} \tag{18}$$

Thus, putting the results in (17) and (18) together, we have

$$(1-\varepsilon)g(\mathcal{S}) \leq f(\mathcal{S}) \leq (1+\varepsilon)g(\mathcal{S}), \tag{19}$$

where $g(\mathcal{S})$ is defined by

$$g(\mathcal{S}) \triangleq \max_{\boldsymbol{r}\in\mathcal{S}^T} \big(s_{r_1,1} + \beta q^0_{r_1} + \sum_{t=2}^{T} (s_{r_t,t} + \beta \bar{q}_{r_t})\big). \tag{20}$$

Notice, however, that $g(\mathcal{S})$ corresponds to the facility location function with modified non-negative similarities defined as $\tilde{s}_{r_1,1} \triangleq s_{r_1,1} + \beta q^0_{r_1}$ and $\tilde{s}_{r_t,t} \triangleq s_{r_t,t} + \beta \bar{q}_{r_t}$ for $t \geq 2$. Hence, $g(\mathcal{S})$ is submodular (Nemhauser et al., 1978; Krause & Golovin, 2014). As a result, from Definition 3, $f(\mathcal{S})$ is ε-approximately submodular. ∎

The level of approximate submodularity, ε, depends on the structure of the transition matrix and the values of transition probabilities. In particular, our utility function becomes closer to submodularity, i.e., ε decreases, when transitions to each given state are more similar. Specifically, we show the following result.

Corollary 1 *Consider the optimization in (9). Assume transition probabilities are such that $\pi(i|1) = \cdots = \pi(i|M)$, for every i. Then, for all $\beta \geq 0$, our utility function in (14) is 0-approximately submodular, i.e., it is submodular.*

Remark 3 *When we have a uniform transition probability from every state, (14) satisfies the condition of the Theorem 1, with $\varepsilon = 0$. In other words, the problem becomes submodular. Notice that in this case, the dynamic potential becomes a constant, hence, has no effect on the solution, and the problem reduces to the standard facility location. Thus, our framework and results, generalize facility location to arbitrary transition probabilities between states.*

The result in Theorem 1 gives a practical algorithm to determine the approximate submodularity level, ε, of the sequential facility location function in (14), for a given transition probability matrix.

Corollary 2 *Consider the optimization in (9) for a given transition model. For log transition probabilities or positive shifted log transition probabilities, define*

$$\ell_i \triangleq \min_{j=1,\ldots,M} q_{j,i}, \quad u_i \triangleq \max_{j=1,\ldots,M} q_{j,i}. \tag{21}$$

The approximate submodularity level of $f(\mathcal{S})$, defined in (14), is given by

$$\varepsilon = \max_{i=1,\ldots,M} \left| \frac{u_i - \ell_i}{u_i + \ell_i} \right|. \tag{22}$$

Proof. The proof follows by setting $\bar{q}_{r_t} = 0.5(u_{r_t} + \ell_{r_t})$ in Theorem 1, for which it is guaranteed that $q_{r_{t-1},r_t} = \bar{q}_{r_t}\psi_{r_{t-1},r_t}$, where $\psi_{r_{t-1},r_t} \in [1-\varepsilon, 1+\varepsilon]$. ∎

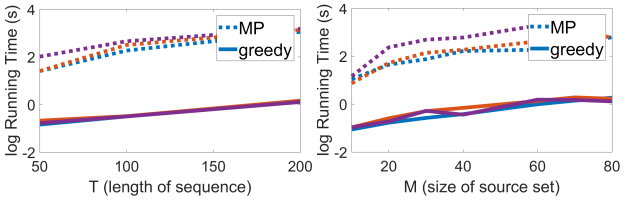


Figure 2: Logarithm of running time (seconds) of our greedy sequential facility location and message passing as a function of (left) the length of the sequence, T, for fixed $M = 50$, (right) the number of states, M, for fixed $T = 200$, for $\beta = 0$ (blue), $\beta = 0.01$ (red) and $\beta = 0.1$ (purple).

4. Experiments

In this section, we evaluate the performance of our method, which we refer to as Greedy Sequential Facility Location (GreedySeqFL), on synthetic data and on the problem of procedure learning from instructional videos.

4.1. Synthetic Experiments

We compare our method with the message passing (MP) algorithm (Elhamifar & De-Paolis-Kaluza, 2017b). We generate synthetic data where the source set $\mathcal{X}$ corresponds to means of M Gaussian distributions with unit variance, and the initial and transition probabilities are generated at random and then normalized. We draw a sequence of length T from the model to form the target set $\mathcal{Y}$ and run different algorithms to select k representatives.

Running Time. We evaluate the running time of both optimization methods as a function of the length of sequences, T and the number of states, M for selecting $k = 10$ representatives. Given that the MP has cubic and quadratic complexity in M and T, respectively, we had to limit $M \leq 80$ and $T \leq 200$. Also, since MP cannot enforce the exact number of representatives, we run MP for a large number of λ's and use the run that achieves k representatives. Figure 2 shows the running times on the logarithmic scale for three values of $\beta \in \{0, 0.01, 0.1\}$. Notice that, for a fixed M and varying T, our greedy method is three orders of magnitude faster than MP, while for a fixed T and varying M, the greedy is two orders of magnitude faster. This is because, while each iteration of MP is two orders of magnitude more costly than the entire run of the greedy, empirically, the number of iterations of MP to converge increases significantly as T increases. Also, the running time of our greedy method does not change with β, whereas MP becomes slower, requiring more iterations to converge, as β increases.

Objective Value. We compare the achieved value of the objective function in (9) for both our method and MP, over 100 random trials, as a function of the number of representatives. Figure 3 illustrates the results for $\beta \in \{0, 0.5, 1\}$ on sequences with $M = 50$ and $T = 200$. While for $\beta = 0$ both methods perform similarly, as β increases, the greedy

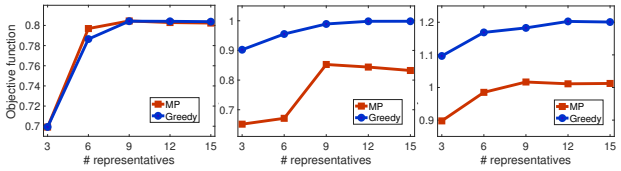


Figure 3: Average objective function value over 100 random trials as a function of the number of representatives selected by our greedy sequential facility location and message passing for (left) $\beta = 0.0$, (middle) $\beta = 0.5$, (right) $\beta = 1.0$.

method achieves higher objective values than MP. Thus, our method performs better than MP in maximizing the utility function, while being several orders of magnitude faster.

Effect of ε. Finally, we investigate the effect of the approximate submodularity level of our utility function on the performance of the greedy method. To do so, we generate transition matrices for which our formulation in (9) is ε-approximately submodular, hence, the condition in (22) is satisfied. From (22), we have $\min_j \pi(i|j) \geq (\max_j \pi(i|j))^{(1+\varepsilon)/(1-\varepsilon)}$ for every i. We assume that each row of the transition matrix contains t probabilities corresponding to $\alpha_u \triangleq \max_j \pi(i|j)$ and the remaining $M-t$ probabilities correspond to $\alpha_\ell \triangleq \min_j \pi(i|j)$. Our goal is to find α_u and α_ℓ of the largest difference $\alpha_u - \alpha_\ell$ that satisfy (22) and give a valid transition probability matrix. Thus, we solve

$$\max_{\alpha_u, \alpha_\ell} \alpha_u - \alpha_\ell, \ \text{s.t.} \ \alpha_\ell \geq (\alpha_u)^{\frac{1+\varepsilon}{1-\varepsilon}}, \ t\alpha_u + (M-t)\alpha_\ell = 1. \tag{23}$$

Once we obtain the solution[2], we randomly select t entries of each row of the transition matrix and set the their values to α_u and set the remaining entries to α_ℓ. Thus, each column consists of α_u and α_ℓ values, satisfying the condition for ε-approximate submodularity. To investigate the effectiveness of our method, we also perform exhaustive search to compute the global maximum of (9), which we refer to as Oracle. Figure 4 shows the average objective value, over 10 trials, as a function of ε and for $M = 30, T = 1000$ and $k \in \{3, 4\}$ representatives (we could not run Oracle for larger values of M or k, due to prohibitive computational cost). Notice that when the problem is exactly submodular, i.e., $\varepsilon = 0$, the gap between the two methods is smaller and it grows as soon as the problem becomes approximately submodular, i.e., $\varepsilon > 0$. However, the gap in achieved objective values remains relatively the same for all values of $\varepsilon \in [0.1, 0.9]$. This shows that the performance of the greedy is in practice much better that the theoretical guarantee, which is only for the worst-case scenario.

4.2. Real Experiments on Procedure Learning

We evaluate the performance of our proposed method for procedure learning from instructional videos (Alayrac et al., 2016; Sener & Yao, 2018). Assume we have a collection of L instructional videos, $\mathcal{Y}_1, \ldots, \mathcal{Y}_L$, of the same task, from which we want to learn a procedure, i.e., key steps and their ordering to achieve the task. Despite variations in videos, such as visual appearance, view points and length of videos, one can identify key segments, where the action in each segment is seen in many videos, as well as an ordering in the sequence of key segments, common across most videos.

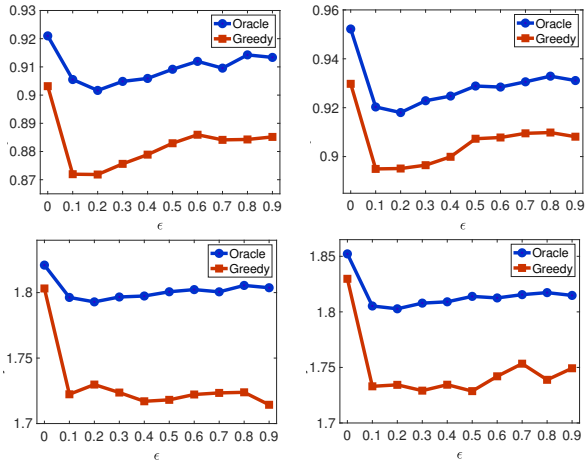


Figure 4: Average objective function value, over 10 trails, for $\beta = 0.1$ (top) and $\beta = 1$ (bottom) as a function of the approximate submodularity level ε for $k = 3$ (left) and $k = 4$ (right).

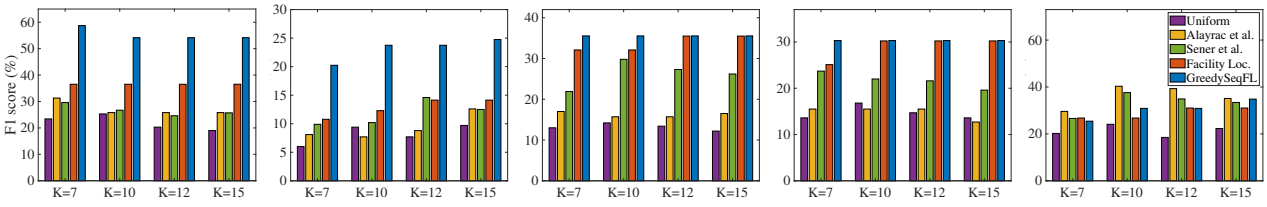
We use our proposed method to recover the common sequence of key steps as the procedure description and localize them in all videos. To do so, we segment each video using (Gygli et al., 2014) and, following (Alayrac et al., 2016), extract a 3000-dimensional feature vector from each segment, capturing appearance and motion, reduce the dimension of the data to d via PCA, hence, obtaining a time-series representation $\mathcal{Y}_\ell$ for each video ℓ. We then learn, from all input videos, an HMM, whose states gathered in $\mathcal{X}$ correspond to different sub-activities across videos. In addition, the transitions between states in π capture the ordering of the steps across videos. Given that key-steps are common across many videos and their ordering must follow the dynamic model (π_0, π), we apply our proposed method to select a representative subset of states from $\mathcal{X}$. Once we recover the sequence of representatives for each video, we align them using multiple sequence alignment techniques (Wang & Jiang, 1994; Alayrac et al., 2016) to produce a compact procedure description of length K. Finally, to localize key-steps, we find all segments in all videos that are assigned to a key-state in the output of our optimization.

We perform experiments on the Inria instructional video dataset (Alayrac et al., 2016) that consists of five tasks of 'change tire', 'make coffee', 'perform cpr', 'jump-start car' and 'repot plant', with 30 videos per task. We compare our framework with Alayrac et al. (Alayrac et al., 2016),

[2] We use CVX to solve (23) and check the solution feasibility.

	Uniform	Alayrac et al.	Sener et al.	Facility Loc.	GreedySeqFL
$K = 7$	15.2	20.3	22.3	26.3	**34.1**
$K = 10$	18.0	21.0	25.3	27.6	**34.9**
$K = 12$	14.8	21.0	24.6	29.5	**34.9**
$K = 15$	15.4	20.5	23.5	29.5	**35.9**

Table 1: Average F1 scores (%) of different algorithms on the Inria Instructional dataset for different values of procedure length, K.



(a) perform cpr (b) jump-start car (c) make coffee (d) repot plant (e) change tire

Figure 5: F1 scores (%) of different algorithms on the Inria Instructional dataset for different values of procedure length, K.

Sener et al. (Sener & Yao, 2018) and a baseline, referred to as Uniform, where we distribute key-step assignments uniformly over all segments in each video. In addition, we consider the Facility Location method in which no dynamic model is used. Specifically, instead of fitting an HMM, we perform Kmeans on feature vectors of segments of all videos from the same task, followed by running Facility Location, i.e., our optimization in (5) with $\beta = 0$. This allows us to investigate the effectiveness of using the dynamics of data for subset selection and procedure learning.

For the experiments, we set $d = 300$, $M = 50$, $k = 15$ and $\beta = 0.01$ in our method. To evaluate the results, we compute the F1-score, which is the harmonic mean of the precision and recall and takes a value between 0 and 1. The precision is the ratio of the total number of correctly localized key-steps to the total number of discovered key-steps across videos. Similarly, the recall would be the ratio of the total number of correctly localized key-steps to the total number of ground truth key-steps across videos.

Results. Table 1 shows the average F1 scores of different algorithms on the Inria dataset as a function of the final procedure length $K \in \{7, 10, 12, 15\}$. Notice that our method performs significantly better than other methods for all values of K. In particular, we achieve the F1 score of 34.1% for $K = 7$ compared to 22.3% via Sener et al. and achieve the score of 35.9% for $K = 15$ compared to 23.5% via Sener et al. On the other hand, both Facility Location and our method outperform other algorithms, demonstrating the effectiveness of subset selection for addressing procedure learning. However, our method improves over Facility Location by at least 5.4%, which shows the importance and effectiveness of incorporating the dynamic model of data into subset selection.

Figure 5 show the results of different algorithms for each of the 5 tasks. Notice that, except for the task of 'change tire', our method outperforms other methods across all values of K. Alayrac et al. perform better for 'change tire' as it takes advantage of both speech and visual data, where speech seems to be helpful for this task. Our method obtains its best performance on the task of 'perform cpr', achieving F1 scores of 58.7% for $K = 7$ compared to 31.3% and 36.5% by Alayrac et al. and Facility Location, respectively. This is due to the fact that in this task, one has to alternate between the two steps of 'give compression' and 'give breath' multiple times, hence, taking advantage of the dynamic model of data becomes important.

Figure 6 shows the effect of β on the performance of our method on three tasks. Here, we show the improvement with respect to $\beta = 0$. Notice that the performance improves as β increases from zero, with 'perform cpr' enjoying the largest improvement in the F1 score. On the other hand, when β becomes sufficiently large, the performance of 'make coffee' decreases with respect to $\beta = 0$, as we overemphasize on the dynamic potential and ignore the encoding.

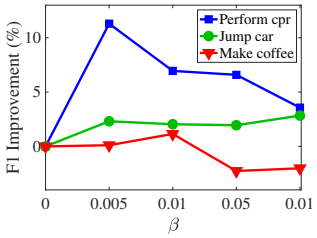


Figure 6: F1 score improvement with respect to $\beta = 0$.

5. Conclusions

We proposed a utility function and a fast greedy algorithm for subset selection in sequential datasets, taking advantage of the dynamic model of data. We proved that under appropriate conditions on transition dynamics, our utility function is ε-approximately submodular, hence, enjoys approximate guarantees via the greedy method. By experiments on synthetic and real data, we showed the effectiveness of our method in terms of running time and attained objective values as well as addressing the procedure learning task.

Acknowledgements

This work is partially supported by grants from NSF (IIS-1657197), DARPA Young Faculty Award (D18AP00050), ONR (N000141812132) and ARO (W911NF1810300).

References

Affandi, R. H., Kulesza, A., and Fox, E. B. Markov determinantal point processes. In *Conference on Uncertainty in Artificial Intelligence*, 2012.

Alayrac, J. B., Bojanowski, P., Agrawal, N., Laptev, I., Sivic, J., and Lacoste-Julien, S. Unsupervised learning from narrated instruction videos. 2016.

Awasthi, P., Bandeira, A. S., Charikar, M., Krishnaswamy, R., Villar, S., and Ward, R. Relax, no need to round: Integrality of clustering formulations. In *Conference on Innovations in Theoretical Computer Science (ITCS)*, 2015.

Bellman, R. E. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 2003.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. In *International Conference on Machine Learning*, 2017.

Borodin, A. and Olshanski, G. Distributions on partitions, point processes, and the hypergeometric kernel. *Communications in Mathematical Physics*, 211, 2000.

Calinescu, G., Chekuri, C., Pal, M., and Vondrak, J. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 40, 2011.

Carbonell, J. and Goldstein, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

Civril, A. and Magdon-Ismail, M. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410, 2009.

Elhamifar, E. and De-Paolis-Kaluza, M. C. Online summarization via submodular and convex optimization. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Elhamifar, E. and De-Paolis-Kaluza, M. C. Subset selection and summarization in sequential data. *Neural Information Processing Systems*, 2017b.

Elhamifar, E., Sapiro, G., and Vidal, R. See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012a.

Elhamifar, E., Sapiro, G., and Vidal, R. Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. *Neural Information Processing Systems*, 2012b.

Elhamifar, E., Burden, S., and Sastry, S. S. Adaptive piecewise-affine inverse modeling of hybrid dynamical systems. In *World Congress of the International Federation of Automatic Control*, 2014.

Elhamifar, E., Sapiro, G., and Sastry, S. S. Dissimilarity-based sparse subset selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

Esser, E., Moller, M., Osher, S., Sapiro, G., and Xin, J. A convex model for non-negative matrix factorization and dimensionality reduction on physical space. *IEEE Transactions on Image Processing*, 21(7):3239–3252, 2012.

Feige, U. A threshold of ln n for approximating set cover. *Journal of the ACM*, 1998.

Feldman, M., Naor, J., and Schwartz, R. A unified continuous greedy algorithm for submodular maximization. *Symposium on Foundations of Computer Science*, 2011.

Frey, B. J. and Dueck, D. Clustering by passing messages between data points. *Science*, 315, 2007.

Gong, B., Chao, W., Grauman, K., and Sha, F. Diverse sequential subset selection for supervised video summarization. In *Neural Information Processing Systems*, 2014.

Gonzalez, T. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 1985.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003.

Gygli, M., Grabner, H., Riemenschneider, H., and Gool, L. V. Creating summaries from user videos. In *European Conference on Computer Vision*, 2014.

Hadlock, F. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4, 1975.

Hartline, J., Mirrokni, V. S., and Sundararajan, M. Optimal marketing strategies over social networks. In *World Wide Web Conference*, 2008.

Horel, T. and Singer, Y. Maximizing approximately submodular functions. In *Neural Information Processing Systems*, 2016.

Joshi, S. and Boyd, S. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57, 2009.

Kim, G., Xing, E., Fei-Fei, L., and Kanade, T. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *International Conference on Computer Vision*, 2011.

Krause, A. and Golovin, D. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014.

Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *Journal of Machine Learning Research*, 9, 2008.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5, 2012.

Lin, H. and Bilmes, J. Learning mixtures of submodular shells with application to document summarization. In *Conference on Uncertainty in Artificial Intelligence*, 2012.

McSherry, D. Diversity-conscious retrieval. In *Advances in Case-Based Reasoning*, 2002.

Mirchandani, P. B. and Francis, R. L. *Discrete Location Theory*. Wiley, 1990.

Misra, I., Shrivastava, A., and Hebert, M. Data-driven exemplar model selection. In *Winter Conference on Applications of Computer Vision*, 2014.

Motwani, R. and Raghavan, P. Randomized algorithms. *Cambridge University Press, New York*, 1995.

Nellore, A. and Ward, R. Recovery guarantees for exemplar-based clustering. In *Information and Computation*, 2015.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

Sener, F. and Yao, A. Unsupervised learning and segmentation of complex activities from video. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Shah, A. and Ghahramani, Z. Determinantal clustering process – a nonparametric bayesian approach to kernel based semi-supervised clustering. In *Conference on Uncertainty in Artificial Intelligence*, 2013.

Simon, I., Snavely, N., and Seitz, S. M. Scene summarization for online image collections. In *IEEE International Conference on Computer Vision*, 2007.

Tschiatschek, S., Singla, A., and Krause, A. Selecting sequences of items via submodular maximization. *AAAI*, 2017.

Wang, L. and Jiang, T. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4), 1994.