simple and efficient greedy algorithm performs near-optimally with approximation factor $1 - 1/e$ [36]. In practice, moreover, as the approximation factor of the greedy algorithm on submodular maximization is usually very close to 1 [27], one could expect very little additional risk when using Algorithm 1 with approximate inference on learning submodular shell mixtures.

To the best of our knowledge, Theorem 1 is the first approximate learning bound for subgradient algorithms with undergenerating (greedy) inference.

### 5 Application to Document Summarization

Submodular mixtures could be applied to many structured prediction problems of practical interest. In this paper, we apply submodular shell mixture learning to extractive document summarization as a case study.

#### 5.1 Submodularity in Document Summarization

Extractive document summarization can be seen as a subset selection problem [27]. Given a ground set of sentences $V$, the task of extractive document summarization is selecting a subset of sentences, say $S$, that best represents the whole document. In other words, we want to find $A \subseteq V$ such that

$$A \in \arg\max_{B \subseteq V} f(B) \text{ subject to: } \sum_{i \in B} c_i \leq b,$$

where $c_i \in \mathbb{R}^+$ is the cost of sentence $i$ (e.g., it could be the number of words in the sentence), $b \in \mathbb{R}^+$ is the total budget (e.g., it could be the largest number of words allowed in a summary), and $f : 2^V \to \mathbb{R}$ is a set function that models the quality of a summary. Eqn (4) is known as the problem of submodular maximization subject to knapsack constraints [31] which NP-complete [35]. However, when $f$ is monotone submodular, Eqn (4) can be solved efficiently and near-optimally with a theoretical guarantee via greedy algorithms [48, 27].

One can always force $f$ to be submodular, leading to an objective function that can be optimized well but might on the other hand poorly represent a given problem. One attractive property of submodularity, like convexity in continuous domain, is that it arises naturally in many applications. One such applications is document summarization. As pointed out in [28], many well-established methods, including the widely used maximum margin relevance method [3], actually correspond to submodular optimization. Moreover, it is shown that the commonly used ROUGE score [26] for automatic summarization evaluation is monotone submodular [28], giving further evidence that submodular functions are natural for document summarization.

In this paper, we further show that not only is the ROUGE score submodular, the score used in the Pyramid method [37], one of the manual evaluation metrics that has been used in recent TAC summarization track\(^2\), is also monotone submodular.

**Theorem 2.** The modified score in Pyramid method is monotone submodular.

The proof is in Appendix C in the supplement.

The remaining question is how to design (or ideally learn) a good submodular function for summarization. Lin and Bilmes [28] proposed a class of submodular functions that models the coverage as well as the diversity of summary. In this paper, we further generalize their class of submodular functions and propose to use submodular shell mixtures for document summarization.

#### 5.2 Submodular shells for summarization

**Diversity shell components**

We define a diversity shell component as

$$f_{\text{diversity}}(a, K, A, (V, \beta))(S) = \frac{\sum_{k=1}^{K} (\sum_{i \in S \cap P_k} r_i)^a}{\sum_{k=1}^{K} (\sum_{i \in P_k} r_i)^a},$$

where $0 \leq a \leq 1$ is the curvature, $K \in \mathbb{Z}^+$ is the number of clusters (partitions), $A$ is a clustering algorithm, and $\{P_k\}_{k=1}^{\cdots, K}$ is a partition of the ground set $V$ generated by $A$, and $r = \{r_v\}_{v=1}^{V}$ with $r_i \in [0, 1]$ is the vector of singleton reward of element $i \in V$. The diversity component models the diversity of a summary set $S$, by diminishing the benefit of choosing elements from the same cluster.

Note that the $\alpha$ parameter of a submodular shell here takes the form $(a, K, A)$. By using different values of $a$ and $K$, and different clustering algorithms $A$, we can produce a variety of submodular shells. The $(V, \beta)$ parameter of a submodular shell takes the form of $(V, r)$. When a document (ground set) is given, rewards of each sentence (i.e., $r_i$) can be computed, and the diversity shell component is then instantiated into a submodular function that measures the diversity of a summary for this particular document.

**Clustered facility location shell components**

We define clustered facility-location like components as

$$f_{\text{c-facility}}(K, A, (V, \beta))(S) = \frac{1}{K} \sum_{k=1}^{K} \max_{i \in S \cap P_k} r_i,$$