Visual Analysis of Submodular Point and Feature Selection for Data-Efficient Machine Learning

Paul Trust, Haseeb Younis, Ahmed Zahran, and Rosane Minghim

Abstract—As data grows exponentially in today's big data landscape, the computational challenge of analyzing and visualizing vast datasets, often with millions of records, becomes increasingly difficult. While modern machine learning models, particularly those based on deep learning, utilize this data to achieve state-of-the-art performance across various tasks and applications, gaining insights through visualization is nearly infeasible. Deep learning's notable performance often comes at the expense of training on high-quality, massive datasets, necessitating multiple runs for optimal hyperparameters. However, acquiring such datasets is not trivial, as the laborious labeling process requires domain expertise and increases both computational resource costs and carbon emissions. In this paper, we investigate the use of carefully selected data subsets, including points or features, to represent larger datasets using submodular data selection functions. These representative summaries facilitate efficient training of machine learning models and enable visualization. We perform an in-depth visual analysis of the feature space chosen by these algorithms, assessing not only accuracy but also cluster and manifold structure preservation. We propose a two-stage subset selection process that combines clustering and importance sampling to induce diversity within subsets and reduce the computational requirements of submodular functions. Moreover, we present SUB-SELECT, a visualization tool that allows users to perform point selection on large datasets and provides a visual comparison by employing multidimensional projections of the selected points. Our experimental results on both text and vision datasets demonstrate that a carefully selected 10% of the data can retain most of the interesting statistical properties of the original dataset, making this approach more accessible and computationally feasible while maintaining the quality of machine learning models.

Index Terms—Data Selection, Submodularity, Multidimensional Projections

1 Introduction

In modern times, digital data is prevalent, and a vast amount of data is generated from various sources, such as social media, sensors, financial data, medical records, and others. This accumulation of data, known as "Big Data," has become an area of interest for academics, businesses, and governments. The growth of data is due to factors like the internet of things, social media, and digitization of offline records. However, the large size and high dimensionality of these datasets present challenges to both machine learning and visualization communities, making it difficult to analyze and visualize them efficiently.

Machine learning systems, particularly deep learning models, have utilized big datasets to achieve state-of-the-art performance in various tasks and applications. However, this impressive performance relies heavily on expensive computational resources and high-quality labeled data. Although deep learning reduces the need for feature engineering through automatic feature extraction, it still faces challenges such as expensive hyper-parameter tuning, which requires multiple runs on the entire dataset, increasing computational and resource costs [33], end-to-end training, and carbon emissions [33]. Additionally, the large volume of data creates new challenges, such as gathering, storing, analyzing, and visualizing. The training data often contains irrelevant and redundant features, which provide no additional predictive advantage and increase computational resources.

One way to alleviate the high dimensionality and large size of the datasets is through feature and point selection. Reducing the feature space is often known as dimensionality reduction. Dimensionality reduction algorithms can be categorized into feature extraction or feature selection. Feature extraction techniques transform the high-dimensional datasets into lower-dimensional datasets by preserving the underlying structure of the original data. Examples include principal components analysis [21], Least Square projection (LSP) [27], t-stochastic Neighborhood Embedding (t-sne) [3] among others. Feature selection on the other hand involves reducing the input features to the most informative ones for use in model construction. Feature selection methods can be classified into filter, wrapper and embedded methods. Filter methods evaluate and select features prior to the learning process. Wrapper methods use the performance of the learning algorithms that will be deployed on the selected features to select the most important features. Wrapper methods are often more accurate compared to filter methods but are much more computationally expensive. Embedded methods leverage the strength of filter methods and wrapper methods. They start by using a statistical criteria to select a subset of features just like in filter methods and then uses the predictive accuracy of the learning algorithm to select those features with the best predictive accuracy. Embedded methods achieve better accuracy compared to both filter and wrapper models. Some of the traditional feature selection algorithms commonly used include Lasso [34], Laplacian score [7], relief [30] and mutual information [4].

Point selection, on the other hand selects subsets of representative points to be used for either training machine learning models or visualization. Common approaches to point selection include sub-modular functions, core sets, meta-heuristics, knockoffs, Bayesian optimization, random selection, and determinantal point processes. The modern methods not only aim to select the most representative subsets but enjoy theoretical guarantees on the selected subsets the convergence rates and negligible loss in predictive accuracy on test sets. The goal of point selection is to train a machine learning model on small subsets of large datasets with minimal or negligible loss and performance in the test set [12].

In this paper, we address the following question: *Given limited resources (time and budget), how can we optimally select a subset (features or points) of a larger dataset for visualization and training machine learning models in an efficient way without significant loss in performance and interesting statistical properties of the larger dataset.* Our proposed approach is based on the submodular functions. Submod-

- *Paul Trust is with University College Cork. E-mail: 120222601@umail.ucc.ie*
- *Haseeb Younis is with University College Cork. E-mail: 121126205@umail.ucc.ie .*
- *Ahmed Zahran is with University College Cor E-mail: a.zahran@cs.ucc.ie .*
- *Rosane Minghim is with University College Cor E-mail: r.minghim@cs.ucc.ie .*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

ular functions often used in economics, operations research, and (most recently) machine learning exhibit the property of diminishing returns, a property akin to subset selection. Submodular maximization also generalizes to many well-known problems like maximum weighted matching, and maximum coverage and finds applications in many machine learning use cases such as data selection [37], active learning [9], document summarization [18], image and video collection summarization [11]. Moreover, a seminal result of Nemhauser et al. [26] showed that a simple greedy algorithm for optimizing the submodular function can produce solutions competitive with an optimal (intractable) solution. Most submodular functions involve an expensive operation of computing a similarity matrix among the features. Our contributions are two-fold, one is exploring the potential of subset selection before visualization for the case of high dimensional and large datasets, and also creating a tool for enabling users to explore the visual space selected by submodular functions. Secondly, we propose a two-stage sampling strategy that first clusters the data before performing the submodular selection.

2 Background and Related Work

2.1 Submodular Data Selection

Consider a collection $V = 1,2,3,...,n$ of elements, referred to as the ground set. We define a utility function (set function) $f : 2^V \rightarrow \mathscr{R}$ as a measure of the quality of a subset $X \subseteq V$. Also, let $c : 2^V \rightarrow \mathscr{R}$ be a function representing the cost of the set, for instance, the size of the subset. In many cases, the cost c is subject to budget constraints. A common formulation for this is presented in Equation 1:

$$\max f(X) \text{ subject to } c(X) \leq b \tag{1}$$

Here, the objective is to find a subset X that maximizes f while adhering to the constraint that the size of the set is less than or equal to the budget b. Maximizing a general set function becomes computationally intractable as the size of the set V increases. However, a specific class of set functions called submodular functions [9] simplifies this optimization. A discrete function $f : 2^V \rightarrow \mathscr{R}$, which returns a real value for any subset $S \subseteq V$, is submodular if for every $A \subseteq B \subseteq V$:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{2}$$

Given $g \in V$ and $g \notin B$, an alternative definition of submodularity is demonstrated in Equation 3:

$$f(g \cup A) - f(A) \geq f(g \cup B) - f(B) \tag{3}$$

Equation 3 highlights the diminishing returns property of submodular functions, meaning that adding element g to set B yields a smaller gain in the target function than adding g to a smaller set A. Intuitively, since B is a superset of A and already contains more information, adding g is less beneficial.

A greedy algorithm can be employed to optimize a submodular function when selecting a subset, offering a lower bound performance guarantee within a factor $(1 - 1/e)$ of the optimal solution. In practice, these greedy solutions are often within a factor of 0.98 of the optimal solution [8, 35]. This makes it preferable to frame the objective for data selection as a submodular function.

2.2 Related Work

Common approaches for training data subset selection especially for deep learning is to use a coreset which efficiently approximate various geometric measures over a large set of feature via a small subset [9]. There is an increasing interest of researchers in machine learning to use sub-modular functions to solve machine learning problems. These functions naturally model the notion of diversity, representation, and coverage and submodular function optimization has been widely used in training data subset selection algorithms for deep learning models [9]. Several works [13, 25, 36, 37] have studied submodularity for data selection to enable efficient training of machine learning models. Vishar et.al [9] uses submodular selection functions and active learning methods to select efficiently select training data for computer vision. Jeff et.al [37] select a large set of acoustic data to train automatic speech recognition systems using submodular functions and demonstrate this strategy significantly performs random selection. Katrin and Jeff [14] introduced submodular optimization to the problem of training data subset selection for statistical machine translation. Unlike most of the previous works, our studies the potential of using subsets for efficient visualization of larger datasets.

Feature selection is a widely studied area and there has been some systems that have been proposed for interactive feature selection. Krause et al. [15] proposes a visual analysis system called INFUSE (Interactive Feature Selection) that helps analysts understand how predictive features are being ranked across automated feature selection algorithms. Minghim et al. [23] proposes a similarity-based graph layout to support feature selection and analysis, they construct a similarity tree to summarize information in a complete weighted graph of features and allow the user to select features that reveal interesting patterns. May et al. [20] proposed SmartStripes, an interactive feature selection tool that enables users to investigate the dependencies and interdependence between different features and entity subsets of features and items related to a target feature. Arthur and Minghim [1] proposed a dual radial visual approach that supports correlation analysis of features combined with data analysis. Rauber et al. [28] presents a system that uses visual representations based on multidimensional projections and feature selection for predictive feedback on classification algorithms and improving classification systems. Our work presents a strategy that combines both feature and point selection using submodular data selection functions and also provide an interactive system (SUB-SELECT) that enables domain experts to perform data selection and compare the selected subsets using multidimensional projections.

3 Methodology

This section presents an overview of the methodology implemented in our system for generating visualizations of data subsets. Specifically, we begin by describing how we transform raw data, which may take the form of text or image, into a numerical format suitable for analysis. We then outline our point selection algorithms, which allow us to obtain a subset that accurately represents the entire dataset. Additionally, we detail our visual feature selection process, which eliminates redundant features to ensure optimal visualizations. To be more precise, we are presented with a set of N data points, denoted as $V = \{1,2,3,...,N\}$, which can be represented in various modalities, including numeric, text, or images. Our primary objective is to identify a subset, $X \subseteq V$, that captures important statistical properties such as accuracy, clusters, and manifolds of the original set. Following this, we perform feature selection on the selected subset to further enhance its usefulness for effective and efficient visualizations and explorations utilizing multidimensional projections.

We prepare our datasets for submodular data selection models, by initially converting raw data into a numerical format. For text datasets, we leverage advanced sentence embeddings generated using state-of-the-art sentence transformers [29]. We opt for sentence embeddings as they tend to maintain clusters within the source documents compared to alternative embeddings, such as naive transformer embeddings or bag of words. This aligns with our use case since our ultimate objective is to visualize subsets utilizing multidimensional projections. In the case of images, we extract features using the output of a convolution neural network (CNN).

3.1 Latent space Clustering and Importance Sampling

It has been observed that numerous submodularity functions require the computation of a similarity matrix among the features present in the dataset, which is at least $O(n^{(2)})$ in complexity. As a result, for larger datasets, this computation becomes prohibitively expensive. For datasets with millions of data points, the computational cost immediately becomes computationally unfeasible to fit within a computer's memory, even one with approximately 8 Gigabyte Random Access Memory (RAM). To mitigate this problem, we first group the data points into K clusters via clustering before performing submodular data

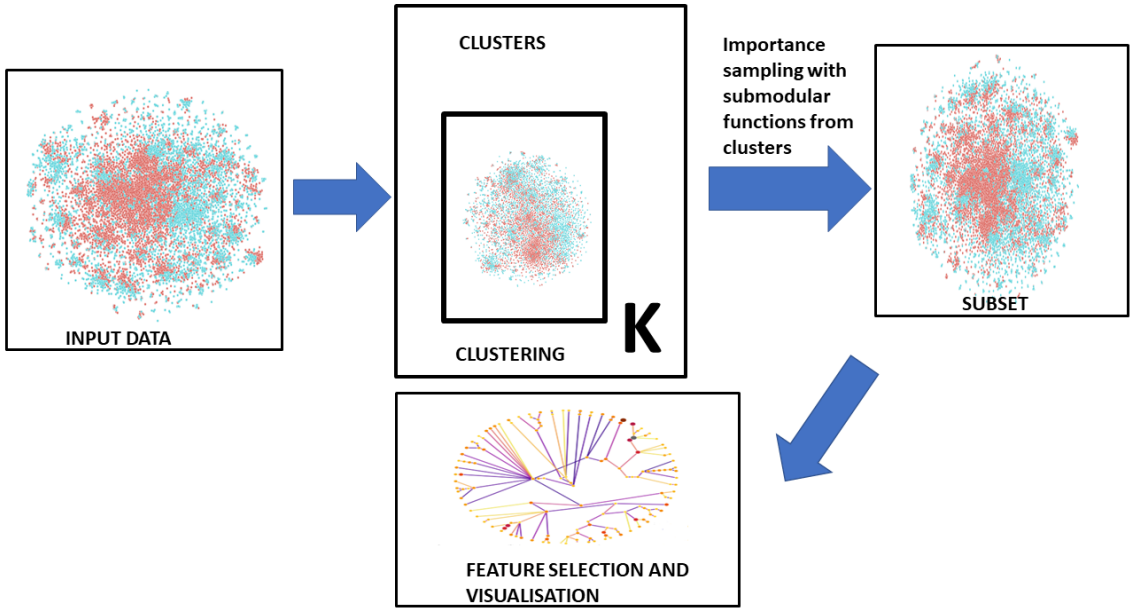


Fig. 1: The input to our system is a large dataset, we first group the data into K clusters and then use submodular data selection to perform importance sampling from each of the clusters to form a final subset. Subset selection is supported by our system SUB-SELECT. The final subset is used for feature selection and interactive visualization as well as for efficient training of deep learning models

selection. This step reduces the memory requirements, making the computation more manageable.

To improve the computational efficiency of submodular functions, rather than providing the entire data matrix X as input, we partition X into K groups. Instead of a straightforward partitioning approach, we propose an intelligent modeling of the latent space in which the data is situated. Specifically, we assume that each data point is linked to a latent space $k \in K$, with the expectation that comparable data points will belong to the same group. To accomplish this, we employ k-means [17] to cluster N items into K groups for simplicity. However, in principle, one could use any state-of-the-art model for latent variable modeling

In order to prevent certain clusters or groups from being overly represented in the final subset, we utilize importance sampling to select groups from the K clusters identified by our clustering algorithm for submodular point selection. Specifically, we define n_k as the number of items in the k-th cluster and set the total number of items selected from a cluster to be $n_k = wN$, where $w = \frac{n_k}{N_k}$. The weight of representation in the final subset is defined as the ratio of the number of items in the cluster to the total number of items in the ground set.

3.2 Submodular Functions as Data Selection Models

The following section outlines the submodular functions employed in our proposed system, which are classified into three distinct categories: Coverage Functions, Diversity Functions, and Representation Functions.

3.2.1 Modeling Representation

Functions based on representation aim to model representations by identifying subsets of items that are most similar to the centroids and medoids in clustering, thereby serving as representative subsets.

- **Facility Location Function**: The facility location function is closely related to k-medoid clustering. If we denote s_{ij} to be the similarity between items in our dataset i and j. We define a facility location data selection function as $f_{FL}(X) = \sum_{i \in V} max_{j \in X} s_{ij}$. This means that for each data item i, we compute the representative item from X which is closest to i, and add to the similarities for all data points [11].
- **Graph Cut Functions**: We define the graph-cut family of functions as $f_{GC}(X) = \sum_{i \in V, j \in X} s_{ij} - \lambda \sum_{i,j \in X} s_{ij}$ where λ governs the trade-off between representation and diversity and s_{ij} measures the similarity [11].

3.2.2 Modeling Coverage

This class of functions models notions of coverage by finding out a subset of the ground set X which covers a set of concepts. The following are some of the representative coverage functions we used in our work.

- **Feature Based Functions**: Feature-based functions operate on feature values directly rather than on a similarity matrix or graph derived from those features. If we denote a data item i via a feature representation q_i. This could be for example transformer embeddings for text or features extracted from the second last layer of a convolution neural network for images. The feature-based function is defined as: $f(X) = \sum_{i \in F} \psi(q_i(X))$ where F is a set of features, $q_i(X) = \sum_{j \in X} q_{ij}$ and q_{ij} is the feature i in data item j. Examples ψ that could be used include square root, logarithms, inverse functions, and many other functions [11, 32].
- **Set Cover Functions**: Denote V as the ground set and let $X \subseteq V$ be a subset of our data. Further denote U to denote a set of concepts covered by X. Each data item $i \in X$ contains a subset $U_i \in U$ set of concepts. We define its set cover evaluation (SC) as $f_{SC}(X) = w(_{i \in X} U_i)$, where w_u denotes the weight of concept u [32].
- **Probabilistic Set Cover**: This is a generalization of the set cover function, to include probabilities p_{iu} for each concept u_i in a data item $i \in X$. Probabilistic set cover is defined as $f_{PSC}(X) = \sum_{u \in C} w_u(1 - p_u(X))$ where C is a set of concepts, w_u is the weight of concept u, $P_u(X) = \prod_{j \in X}(1 - p_{uj})$ and p_{xu} is the probability with which a concept u is covered by element x [10].

3.2.3 Modeling Diversity

Diversity-based functions obtain a subset containing the most diverse points. The goal is to have minimum similarity across elements in the chosen subset by maximizing minimum pairwise distances between elements. The key difference between diversity and representation functions is that while diversity functions only looks at the elements

in the chosen subset, representation functions also worries about the similarity of with the remaining elements in the superset. We describe representative functions in our framework.

- **Dispersion Functions**: The goal of dispersion functions is to have a minimum similarity across elements in the chosen subset by maximizing pairwise distances between elements. More formally, denote d_{ij} as the distance measure between data items i and j in our data set. Define a set function $f(X) = min_{i,j} d_{ij}$. This function is not submodular but can still be efficiently optimized via a greedy algorithm. We observe that maximizing this function leads to a subset with maximal minimum pairwise distances, thereby ensuring that a diverse set of items is selected [11, 32].
- **Determinantal Point Processes (DPP)**:Determinantal point processes (DPP) are elegant probabilistic models of repulsion that arise in quantum physics and random matrix theory. They define a distribution over all subsets of a ground set measuring the negative correlations of the elements in each subset. Let $\mathscr{V} = \{1, .., N\}$ denote a finite ground set containing N data items. A point process $\mathscr{P}$ on a discrete set $\mathscr{V}$ is a probability measure on 2^V (the set of all possible subsets of V). $\mathscr{P}$ is called a determinantal point process if there exists a positive semi-definite matrix L indexed by elements of V such that if $V \sim \mathscr{P}$, we have

$$\mathscr{P}(X;L) = \frac{det(L_i)}{det(L+I)} \qquad \sum_{X \subseteq V} det(L_i) = det(L+I) \tag{4}$$

  where $det(.)$ is the determinant of a matrix; I is the identity matrix; $L \in \mathscr{R}^{n \times n}$ is a positive semi-definite matrix known as $L-$ensemble. L_{ij} is a measure of the correlation between sentences i and j, L_s is a sub matrix of L containing only entries indexed by elements of $X \subseteq V$.

  It turns out the close variant $\log \mathscr{P}(X;L)$ is submodular and can be efficiently optimized via a greedy algorithm. Unlike the Dispersion functions, this requires computing the determinant which is $O(n^3)$ where n is the size of the ground set which may not be computationally feasible on large scale [16].

3.3 Feature Selection

We study the potential of using submodular gains as feature importance measures to be used in feature selection. Since submodular functions are designed to work with points rather than features. We begin by transposing our data matrix in order to select features instead of points. We compute the gain of each feature using submodular functions. The gains are normalized and used as feature importance measures to guide feature selection. Other feature selection models used are Pearson correlation [2] and Extra Trees classifier [6].

3.4 Visualization and Interaction

In order to understand the effect of point selection and feature selection on the original statistical properties of the ground set, we visualized and interacted with the selected space of features and points using graphs from features (GFF) software [24] and SUB-SELECT software which we design for subset selection. We use SUB-SELECT which enables the user to upload a large data set, select different submodular functions, cluster the data, and perform subset selection. The user can visualize the selected subset using multidimensional projection and if they are satisfied with the quality of the projection based on the silhouette coefficient. The selected subset from SUB-SELECT X_s with n rows that represent data instances and d columns that represent features with a target feature each is input to GFF.

From X_s, a transpose X_s^T is obtained and a relevance value is evaluated between each feature and the target as means to reflect similarity using the Pearson correlation coefficient, Extra Trees Classifiers, and submodular selection functions. A complete undirected graph is then constructed having one vertex for each feature and with edge weight $e_w(i, j)$ that represent dissimilarity between each pair of features i and j. The dissimilarity measures used include Euclidean distance, cosine distance, Manhattan distance, Chebyshev distance and Pearson correlation. A tree is constructed to summarize the information in the complete graph using a minimum spanning tree and neighbor-joining.

Interaction with the dataset starts on the layout of the tree with vertices shown as circles whose sizes and colors reflect their relationship with the target feature. The tree can either be displayed in radial layout or a force-base layout may be executed to produce a more compact organization of the vertices. Features may be selected by name or relevance with the target feature. After interacting with the features in the subset X_s, the user selects n_s features that are projected in the n_s-dimensional space defined by restricting X_s to the selected features on 2-Dimensional spaces. We perfom further visualization of the subset of points with the selected features using three multidimensional projection techniques; t-sne [3], LSP [27] and UMAP [22].

4 Experiments and Results

4.1 Data

- **Fashion-MNIST**
  Fashion-MNIST is a dataset comprising of 28×28 grayscale images of 70,000 fashion products from 10 categories, with 7,000 images per category. The training set has 60,000 images and the test set has 10,000 images. Fashion-MNIST shares the same image size, data format and the structure of training and testing splits with the original MNIST.
- **IMDB Review**
  The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has $score \geq 4$ out of 10, and a positive review has a $score \geq 7$ out of 10. No more than 30 reviews are included per movie. The dataset contains 25000 train samples and 25000 test samples [19].

4.2 Assessing Quality of Selected Subsets

We compared the selected subsets of features by different algorithms and full datasets using multidimensional projections to assess preservation of manifold and cluster structure and also using classification evaluation metrics to compare algorithms that results into less degradation in performance with subsets; We t-Stochastic Distributed Embedding(tSNE), Least Squares Projection (LSP) and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) for and performed classification on selected subsets and full datasets using supervised learning classifiers

The projection quality of different subsets of features from multidimensional projections was evaluated using the silhouette coefficient [31] and is computed as

$$S = \frac{1}{n} \sum_{i=1}^{n} \frac{(b_i - a_i)}{max\{a_i, b_i\}} \tag{5}$$

where n is the number of features and for each feature i, a_i is the average distance between all features within the same target and b_i is the minimum average distance between all features in other features of different class. Silhouette coefficient S lie in the interval $[-1, 1]$ with values closer to 1 meaning that the projection is better in terms of cohesion and separability.

We also compared performance of classifiers on subsets and full datasets using precision, accuracy and F1-score metrics defined as follows:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{7}$$

$$f1\_score = \frac{TP}{TP + \frac{1}{2}(FP+FN)} \tag{8}$$

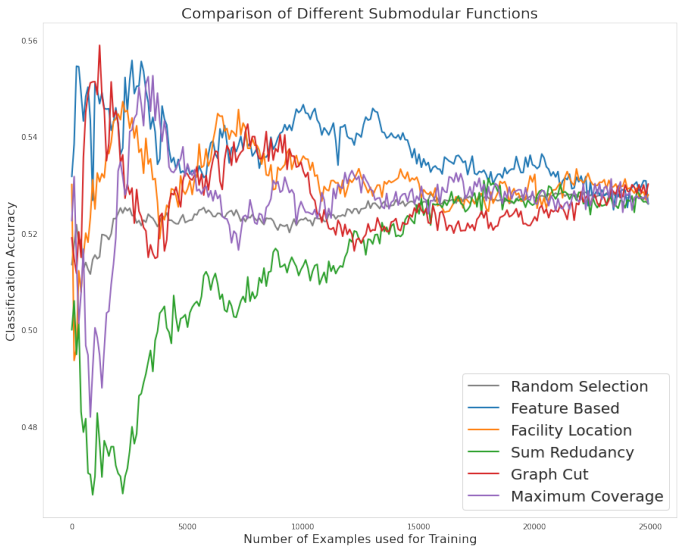


Fig. 2: Classification performance in terms of accuracy for different submodular data selection functions and random selection. The selection are done on IMDB dataset using a simple logistic regression as the classification model

where TP are the True Positives, FP are the False Positives, FN are the False Negatives and TN are the True Negatives.

4.3 Implementation Details

We implement submodular point selection approaches using apricot [32] and submodlib [10] packages. Visualization and interaction with the selected points and features were done using GFF (Graphs from Features) [24] where we added other feature relevance functions based on submodular functions.The classification model for text classification RoBERTa was implemented using huggingface library [38] and text classification for images was implemented in keras [5]. We implement SUB-SELECT, an interactive system that can be used for point selection.

4.4 Does the Selection Procedure matter?

We study how the performance of the subsets (points) selected varies with different submodular data selection functions. Figure 2 demonstrates the classification performance for different subsets based on the logistic regression model. The first observation is that the classification accuracy of logistic regression is much lower compared to the one obtained by the state-of-the-art deep learning model (RoBERTa). We however choose a logistic regression model as a proof of concept since it would be prohibitively expensive to retrain RoBERTa model for multiple times. Observations from Figure 2 demonstrate that there is a noticeable and significant difference in the classification accuracy for different selection functions with most of the submodular data selection functions out-performing random selection for all the selected examples. Such an observation is a motivation for the use of submodular selection since it would be otherwise simple to obtain subsets through a simple random selection. Another key observation is that the performance of selected subsets varies depending on the submodular function used. For example on this particular dataset (IMDB), we observe that feature-based, facility location and graph cut submodular functions consistently outperform other selection functions. However, the difference in performance for different submodular functions may vary depending on the datasets or the task at hand, but one consistent trend is these special submodular functions most of the times out-perform subsets selected with random which provides a justification for submodular selection.

Percentage	Accuracy	f1	Precision	Recall
1	82.8760	82.5571	84.1236	81.0480
5	84.2480	84.7163	82.2705	87.3120
10	85.6280	86.1205	83.2673	89.1760
20	86.4480	86.9521	83.8333	90.3120
30	86.2960	85.7724	89.1796	82.6160
40	87.6160	87.4462	88.6614	86.2640
50	86.7840	87.4886	**92.4160**	86.7940
60	87.2000	87.6760	84.5314	91.0640
70	88.6640	**88.7486**	88.0911	89.0640
80	**88.6960**	88.4935	90.1078	86.9360
90	87.7160	88.2034	84.8371	**91.8480**
100	88.3000	88.2090	88.9006	87.5280

Table 1: Accuracy, f1-score, precision and Recall of submodular point selection based on facility location function for different percentages of the datasets selected and used to train a classification model (RoBERTa) on IMDB dataset. Numbers in bold represent when the best performance was first obtained

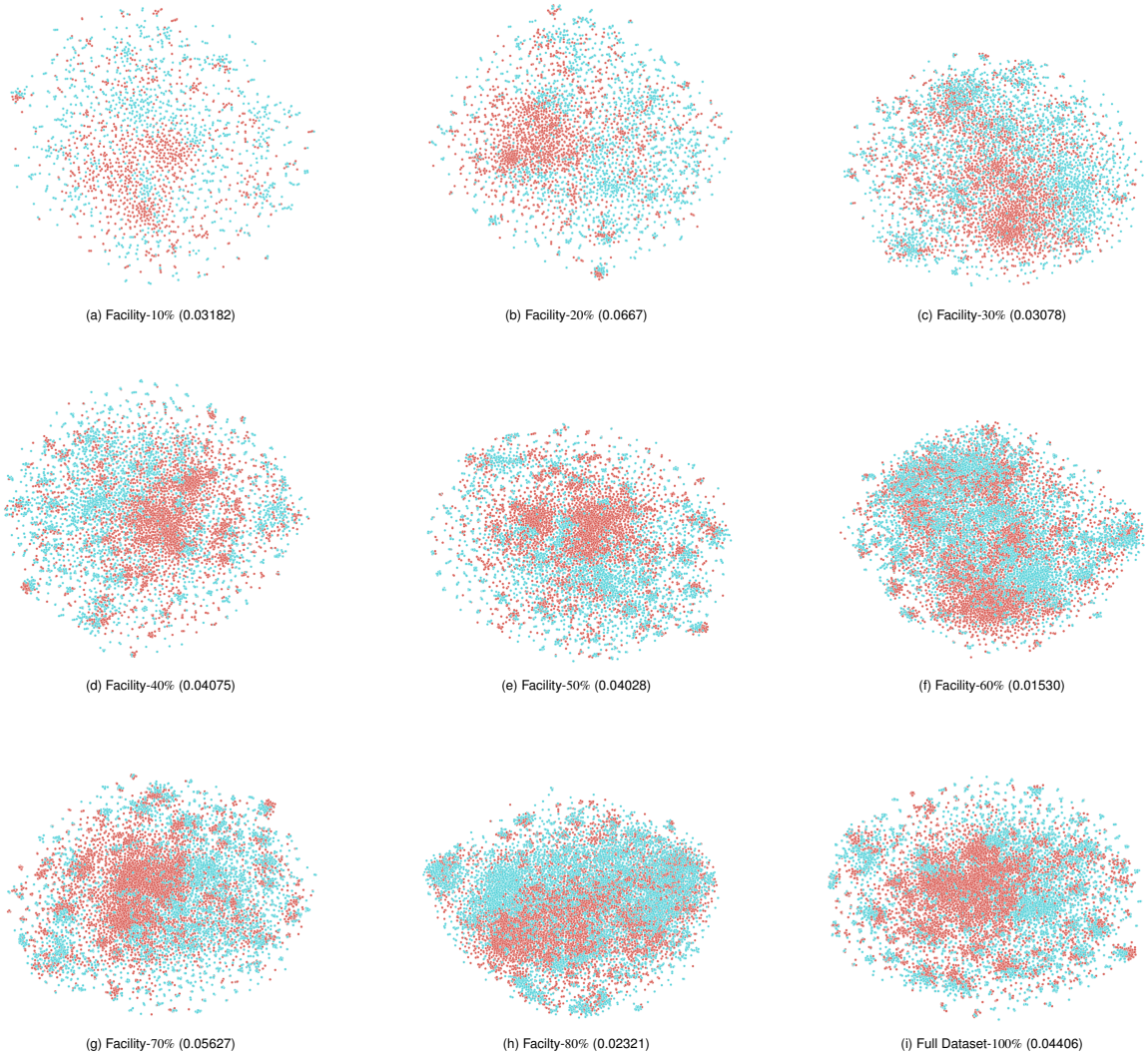
4.5 Classification Results

We performed experiments with the selected subsets for both text dataset (IMDB) and Image dataset (FashionMNIST) to see the effect of submodular selection based on classification performance. The results reported in this paper are based on the facility location submodular data selection function. Even though our end goal is to produce a subset that retains the classification performance of the original large datasets within a certain accepted interval, we perform submodular selection in an unsupervised way without referring to the class labels. This is intuitive since sometimes we may need to select data points for labeling or visualization purposes that may not have labels. Classification for the text dataset was performed with a deep learning model RoBERTa (A Robustly Optimized BERT Pretraining Approach) and classification on the image dataset was performed with a Convolutional neural network.

We present accuracy, f1-score, precision and recall of RoBERTa on IMDB dataset in Table 1, percentages represent a proportion of the dataset that was selected with submodular selection for this particular case based on facility location and used to train a classification model, 100% implies that all the dataset was used in training the model without any selection. The results in Table 1 demonstrate that training the classification model using 80% of the dataset achieves the best accuracy even better than using the full dataset. Such an observation could be attributed to the redundancy that may exist within datasets which sometimes affects performance. We also observe that by only using say 10% of the dataset, we lose a classification performance in terms of accuracy of about only 4% which can be traded for time in some applications. The same trend is observed with f1-score, precision and recall.

We also demonstrate in Figure 4 the difference in classification performance for RoBERTa model based on f1-score between using the subset and full dataset. The graph shows a decreasing trend as we increase the size of the subset. The observation is expected especially for data-hungry models like deep learning models which benefit from an increase in the training data. We however note that the gap in performance between the subset and full set is not very much and drops to about an average below 3% when just using 40% of the original datasets.

Table 2 shows the classification performance of a convolution neural network on FashionMNIST, The findings are not so much different from that one observed with RoBERTa (a text classification model). We observe that using only 10% of carefully selected subsets using submodular functions can yeid an accuracy of about 90.33% which is only about 3% below the desired performance using 100% of the datasets. As earlier observed in the text datasets, the classification performance on subsets can sometimes be better than using a full dataset for example using 70% yields an accuracy of 95.43% which is better than the one obtained using the full training set 92.76%. We obtain the highest f1-score with just 60% of the full dataset. Figure 2 demonstrates



(a) Facility-10% (0.03182) (b) Facility-20% (0.0667) (c) Facility-30% (0.03078)

(d) Facility-40% (0.04075) (e) Facility-50% (0.04028) (f) Facility-60% (0.01530)

(g) Facility-70% (0.05627) (h) Facilty-80% (0.02321) (i) Full Dataset-100% (0.04406)

Fig. 3: t-sne projections of the embedding space of Irish IMDB movie text reviews segregated according to binary categories (positive and negative). Point colors represent categories and the values shown in the parenthesis represent the silhouette coefficient in the projected space. Projections are labeled Facility selection submodular function and the percentage shows the proportion of the original dataset used. The projection on text documents is made on top of sentence embeddings produced by the sentence transformer

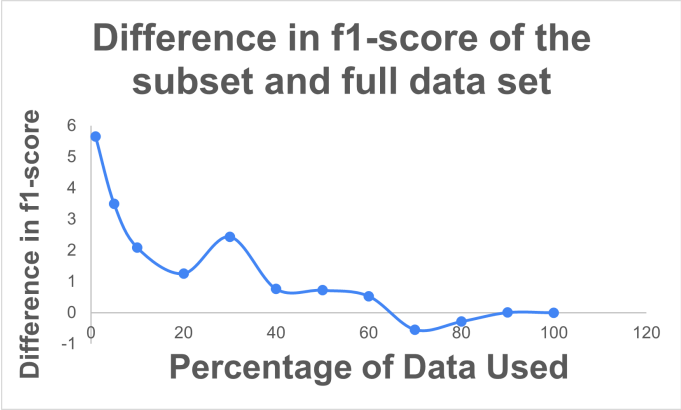


Fig. 4: Difference in f1-score between subsets and full dataset on IMDB dataset selected by submodular data selection based on facility location. The subsets are indicated by the percentage of data used when training the classification model (RoBERTA) with 100% implying the full dataset.

Percentage	Accuracy	f1	Precision	Recall
10	90.33	88.00	88.00	88.00
20	90.95	90.00	90.00	90.00
30	90.45	90.00	91.00	90.00
40	91.92	91.00	91.00	91.00
50	93.86	92.00	92.00	92.00
60	94.89	**93.00**	**93.00**	**93.00**
70	95.43	93.00	93.00	93.00
80	96.25	93.00	93.00	93.00
90	**96.47**	93.00	93.00	93.00
100	92.76	93.00	93.00	93.00

Table 2: Accuracy, f1-score, precision and Recall of submodular point selection based on facility location function for different percentages of the datasets selected and used to train a classification model (CNN) on FashionMNIST dataset. Numbers in bold represent when the best performance was first obtained

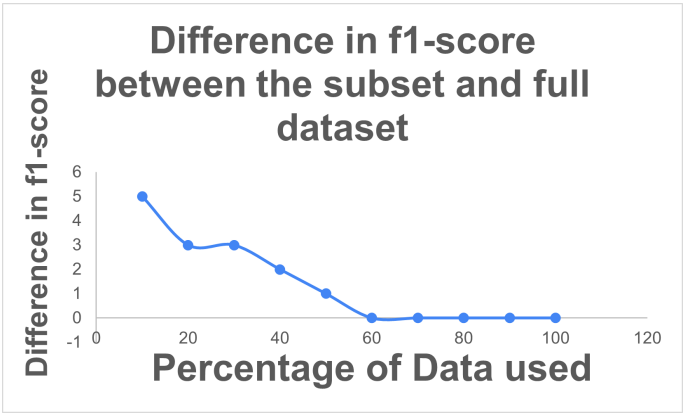


Fig. 5: Difference in f1-score between subsets and full dataset on FashionMNIST dataset selected by submodular data selection based on facility location. The subsets are indicated by the percentage of data used when training the classification model (CNN) with 100% implying the full dataset.

the difference in f1-score for the full dataset and the subset, the graph shows that as we increase the subset, f1-score increases but it reaches the level of f1-score obtained with full dataset with just using 60% of the dataset.

Observations from both text and image dataset demonstrates that deep learning models can benefit from subset selection not only on the reduction on computation power but also on classification performance. We hypothesis that much as the paradigm of deep learning has shown that the more the data, the better the performance, we argue that the quality of data used for training may also be a factor in affecting the classification performance. Data selection offers a promising solution to perform deep learning in an efficient way without very much significant loss in the interesting statistical properties of the original full datasets like f1-score, accuracy, precision and recall. Moreover, we do not need to have labels beforehand when performing submodular data selection which is further advantageous since in most applications it may not be trivial to obtain labeled data.

4.6 Visual Analysis and Exploration of Subsets

To understand how the selected subsets preserve clusters and manifolds for the original datasets, we visualized the projections with multi-dimensional projections. If the end-user wants only to visualize the subset, they can perform subset selection and also visualize at the same time in our tool (SUB-SELECT). if the user wants to perform feature selection on the selected subsets, they can download the subset and use GFF (Graph from Features) [24] tool. [24].

Figure 3 shows t-sne embeddings for IMDB text reviews segregated according to the binary category of whether they are positive or not. The embedding were extracted using sentence transformers [29]. We evaluate the quality of the projections using the silhouette coefficient which is between 0 and 1, the nearer to 1, the better the projection and the closer to 0, the worse the projection. Figure 3i shows the silhouette coefficient of using 100% of the dataset without any selection is 0.04406. Our experiments reveal that a carefully selected subset supported by submodular data selection may result into better-segregated clusters. The difference between the silhouette coefficients of the full datasets and subsets is very small with some subsets having a better silhouette coefficient, for example using 70% has a silhouette of 0.05627 versus 0.04406 of the full datasets. The general observation that we derive from different embeddings for different subsets as shown in Figure 3 is that we can obtain a summary of the data sets that preserves the underlying properties of the original subset if this summary is carefully selected. If we make an assumption that this summary contains the most important information in the ground data set, then it may be sufficient to visualize in order to get an understanding of the full dataset than visualizing the full data set itself. The success of this method

applied to visualization can enable scalable and fast exploration of high-dimensional datasets which would otherwise be computationally infeasible to visualize as a whole.

5 Conclusion

In this study, we have shown that submodular data selection effectively chooses data subsets that maintain the original dataset's intriguing statistical properties, such as accuracy, clusters, and manifolds. However, we acknowledge that submodular properties may not always be scalable to extremely large datasets, as most require the computation of a similarity matrix between features, resulting in a quadratic computational complexity concerning input size. To address this challenge, we propose a two-stage sampling approach: before performing submodular sampling, we first cluster the data into k groups and subsequently sample from these clusters. Our experimental results from classification tasks demonstrate that the selected subsets often maintain classification performance with minimal loss, while simultaneously reducing computational costs. This finding highlights the potential of our proposed approach for effectively managing large datasets in machine learning and visualisation applications.

References

[1] E. Artur and R. Minghim. A novel visual approach for enhanced attribute analysis and selection. *Computers & Graphics*, 84:160–172, 2019. 2

[2] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pp. 1–4. Springer, 2009. 4

[3] K. Bunte, S. Haase, M. Biehl, and T. Villmann. Stochastic neighbor embedding (sne) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012. Advances in artificial neural networks, machine learning, and computational intelligence (ESANN 2011). doi: 10.1016/j.neucom.2012.02.034 1, 4

[4] X. Deng, Y. Li, J. Weng, and J. Zhang. Feature selection for text classification: A review. *Multimedia Tools Appl.*, 78(3):3797–3816, feb 2019. doi: 10.1007/s11042-018-6083-5 1

[5] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022. 5

[6] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006. 4

[7] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *Advances in neural information processing systems*, 18, 2005. 1

[8] V. Kaushal, R. Iyer, K. Doctor, A. Sahoo, P. Dubal, S. Kothawade, R. Mahadev, K. Dargan, and G. Ramakrishnan. Demystifying multi-faceted video summarization: tradeoff between diversity, representation, coverage and importance. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 452–461. IEEE, 2019. 2

[9] V. Kaushal, R. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1289–1299. IEEE, 2019. 2

[10] V. Kaushal, G. Ramakrishnan, and R. Iyer. Submodlib: A submodular optimization library. *arXiv preprint arXiv:2202.10680*, 2022. 3, 5

[11] V. Kaushal, S. Subramanian, S. Kothawade, R. Iyer, and G. Ramakrishnan. A framework towards domain specific video summarization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 666–675. IEEE, 2019. 2, 3, 4

[12] K. Killamsetty, D. S, G. Ramakrishnan, A. De, and R. Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In M. Meila and T. Zhang, eds., *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 18–24 Jul 2021. 1

[13] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, R. I. U. of Texas at Dallas, I. I. of Technology Bombay Institution One, and I. Two. Glister: Generalization based data subset selection for efficient and robust learning. In *AAAI*, 2021. 2

[14] K. Kirchhoff and J. Bilmes. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 131–141. Association for Computational Linguistics, Doha, Qatar, Oct. 2014. doi: 10.3115/v1/D14-1014 2

[15] J. Krause, A. Perer, and E. Bertini. Infuse: interactive feature selection for predictive modeling of high dimensional data. *IEEE transactions on visualization and computer graphics*, 20(12):1614–1623, 2014. 2

[16] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. 4

[17] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003. 3

[18] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 510–520, 2011. 2

[19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA, June 2011. 4

[20] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 111–120. IEEE, 2011. 2

[21] A. Maćkiewicz and W. Ratajczak. Principal components analysis (pca). *Computers and Geosciences*, 19(3):303–342, 1993. doi: 10.1016/0098-3004(93)90090-R 1

[22] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 4

[23] R. Minghim, L. Huancapaza, E. Artur, G. P. Telles, and I. V. Belizario. Graphs from features: Tree-based graph layout for feature analysis. *Algorithms*, 13(11), 2020. doi: 10.3390/a13110302 2

[24] R. Minghim, L. Huancapaza, E. Artur, G. P. Telles, and I. V. Belizario. Graphs from features: Tree-based graph layout for feature analysis. *Algorithms*, 13(11), 2020. doi: 10.3390/a13110302 4, 5, 7

[25] B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020. 2

[26] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978. 2

[27] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008. doi: 10.1109/TVCG.2007.70443 1, 4

[28] P. E. Rauber, A. X. Falcao, and A. C. Telea. Projections as visual aids for classification system design. *Information Visualization*, 17(4):282–305, 2018. 2

[29] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China, Nov. 2019. doi: 10.18653/v1/D19-1410 2, 7

[30] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2004. 1

[31] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 4

[32] J. M. Schreiber, J. A. Bilmes, and W. S. Noble. apricot: Submodular selection for data summarization in python. *J. Mach. Learn. Res.*, 21:161–1, 2020. 3, 4, 5

[33] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, nov 2020. doi: 10.1145/3381831 1

[34] H. S. Shon, K.-S. Yang, C. W. Yoo, and K. H. Ryu. Feature selection method using wf-lasso for gene expression data analysis. In *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '11, p. 522–524. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/2147805.2147889 1

[35] S. Tschiatschek, R. K. Iyer, H. Wei, and J. A. Bilmes. Learning mixtures of submodular functions for image collection summarization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds., *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014. 2

[36] K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In F. Bach and D. Blei, eds., *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, pp. 1954–1963. PMLR, Lille, France, 07–09 Jul 2015. 2

[37] K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3311–3315, 2014. doi: 10.1109/ICASSP.2014.6854213 2

[38] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 5