Submodularity in Data Subset Selection and Active Learning: Extended Version

Kai Wei KAIWEI@U.WASHINGTON.EDU
Rishabh Iyer RKIYER@U.WASHINGTON.EDU
Jeff Bilmes BILMES@U.WASHINGTON.EDU
University of Washington, Seattle, WA 98195, USA

Abstract

We study the problem of selecting a subset of big data to train a classifier while incurring minimal performance loss. We show the connection of submodularity to the data likelihood functions for Naïve Bayes (NB) and Nearest Neighbor (NN) classifiers, and formulate the data subset selection problems for these classifiers as constrained submodular maximization. Furthermore, we apply this framework to active learning and propose a novel scheme called *filtered active submodular selection* (FASS), where we combine the uncertainty sampling method with a submodular data subset selection framework. We extensively evaluate the proposed framework on text categorization and handwritten digit recognition tasks with four different classifiers, including deep neural network (DNN) based classifiers. Empirical results indicate that the proposed framework yields significant improvement over the state-of-the-art algorithms on all classifiers.

1 Introduction

A relatively recent turn of events is that data sets for training machine learning systems are getting extremely large — the unprecedented amount of data at our disposal on which we can train our systems is one of the pillars of machine learning's recent successes. Big data has its own problems, however, namely that it is computationally demanding, resource hungry, and often redundant. One solution to this latter problem is to carefully choose a subset of the data so as to minimize any significant loss in performance. In the context of machine learning training data subset selection, we call this problem *supervised data subset selection* (when training data labels are available), or *unsupervised data subset selection* (when not).

One recent class of methods treats the problem as submodular maximization, where appropriate submodular functions are chosen as the surrogate model to measure the utility of each subset for an underlying task (Shinohara, 2014; Zheng et al., 2014; Hoi et al., 2006; Shamaiah et al., 2010; Prasad et al., 2014; Krause et al., 2008; Das & Kempe, 2011; Wei et al., 2014b; Kempe et al., 2003; Gabillon et al., 2013; Chen & Krause, 2013; Reed & Ghahramani, 2013; Singla et al., 2014; Liu et al., 2013; Iyer et al., 2013). While existing work typically shows very good performance, in some cases a mathematical connection can be made between the true objective and a submodular model, a core goal we wish to further advance in the present paper.

Submodular functions, traditionally studied in mathematics, economics, and operations research, are defined as follows: $f : 2^V \rightarrow \mathbb{R}$, returning a real value for any subset $S \subseteq V$, is *submodular* if it satisfies $f(A) + f(B) \geq f(A \cap B) + f(A \cup B), \forall A, B \subseteq V$. An equivalent definition is *diminishing returns*: $f(j|S) \geq f(j|T), \forall S \subseteq T, j \in V \setminus T$, where $f(j|S) \triangleq f(j \cup S) - f(S)$ is the marginal gain of adding an item j to a set S. A function f is monotone non-decreasing if $f(j|S) \geq 0, \forall S \subseteq V, j \in V \setminus S$. We say that f is normalized if $f(\emptyset) = 0$. Submodular functions naturally model notions of information, diversity, and coverage in many applications. Moreover, they can be optimized efficiently by extremely simple algorithms. For example, a greedy algorithm (Nemhauser et al., 1978) ensures that the cardinality constrained submodular maximization problem can be approximated up to a factor of $1 - 1/e$. This is tight unless $P = NP$ (Feige, 1998). Submodularity can be further exploited to accelerate the greedy implementation leading to an algorithm often called *lazy greedy* (Minoux, 1978) with almost linear time complexity.

1.1 Our Contributions

In this paper, we study submodular functions in connection to data subset selection. We first propose an approach to supervised data subset selection by connecting submodularity to likelihood functions of classifiers. Specifically, we express the utility set function for two simple classes of classifiers, the Naïve Bayes (NB) classifier and the Nearest Neighbor (NN) classifier utilizing submodularity. We identify two classes of submodular functions, *Naïve Bayes submodular* and *Nearest Neighbor submodular*, that naturally model maximum likelihood estimates over data subsets

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

for both NB and NN classifiers. Data subset selection is then performed as submodular maximization.

The Naïve Bayes submodular function is a special case of the "feature based" submodular functions that have been recently defined and used in the literature (Wei et al., 2014b; Kirchhoff & Bilmes, 2014), while the Nearest Neighbor submodular function generalizes the well-known class of facility location function (Mirchandani & Francis, 1990) that have also been previously successfully used for subset selection problems (Mirzasoleiman et al., 2013; Iyer & Bilmes, 2013; Zheng et al., 2014).

Supervised data subset selection for the NN classifier, in particular, has great practical importance, since the NN classifier is non-parametric — i.e., the classifier must essentially memorize, and allocate storage for, the entire training set. The complexity of classifying one sample is dependent on the training set size, which can be expensive for large-scale applications. Our supervised data subset selection strategies reduce the training set size, and when the submodular function used to perform the subset selection is matched with the NN classifier, there is little performance loss even though the NN classifier has significantly less data to memorize.

We then extend the data subset selection problem to an active learning setting. We make a distinction between two extreme forms of active learning (Guillory & Bilmes, 2011): 1) *batch active learning*, where there is one round of data selection and the data points are chosen to be labeled without any knowledge of the resulting labels that will be returned, and 2) *adaptive active learning*, where there are many rounds of data selection, each of which selects one data point whose label may be used to select the data point at future rounds. A hybrid multistage scheme we call *mini-batch adaptive active learning* is where in each round a mini-batch of data points are selected to be labeled, and that may inform the determination of future mini-batches.

We propose a novel multi-stage scheme we call *filtered active submodular selection* (FASS) for the mini-batch adaptive active learning. At every round, as more labeled data becomes available, FASS uses an improved approximation to supervised data subset selection. We show how our framework naturally combines the notions of sample *informativeness* and, via submodularity, *representativeness*. We also show how our method is scalable relative to existing active learning techniques that attempt to combine these two notions. We then empirically demonstrate that our methods (both the purely supervised selection methods, and FASS) outperform existing baselines, including in our results a deep neural network (DNN) case.

1.2 Previous Related Work

There are three fundamental problems related to data subset selection. The first is supervised data subset selection (Wei et al., 2014b; 2013; Shinohara, 2014; Tsang et al., 2005), where the selection algorithm has access to the labels of the training data set. The second problem is unsupervised subset selection, where the algorithm does not use the labels for selecting data (Wei et al., 2014c; Har-Peled & Mazumdar, 2004). The third is active learning, where label queries are made on subsets of data (Settles, 2010; Lewis & Gale, 1994; Seung et al., 1992).

A number of authors have studied these versions of data subset selection, and have observed that submodular functions nicely fit this problem (Shinohara, 2014; Zheng et al., 2014; Hoi et al., 2006; Shamaiah et al., 2010; Prasad et al., 2014; Krause et al., 2008; Das & Kempe, 2011; Tschiatschek et al., 2014; Wei et al., 2014b; Kempe et al., 2003; Wei et al., 2014c). In particular, (Wei et al., 2014c; Shinohara, 2014) study the subset selection of speech data for training speech recognition systems in both the supervised and unsupervised setting, and model the utility of a speech training data as the coverage of a set of speech units through submodular functions. Similarly, (Shamaiah et al., 2010) model the utility a subset of sensors as the reduction on the estimation error covariance matrix and show that this utility function is monotone submodular. (Das & Kempe, 2011) analyze the objective of feature subset selection for linear regression and show that it is often approximately submodular. (Hoi et al., 2006) investigates the batch active learning problem for logistic regression, and connect this to submodular optimization. (Guillory & Bilmes, 2011) study the role of submodular functions in subset selection for very general simultaneously active and semi-supervised learning algorithms. Another thread of work (Guillory & Bilmes, 2010; Golovin & Krause, 2010) provides a link between submodularity and active learning through notions of interactive and adaptive submodularity. In much of this related work, the submodular function is heuristically chosen to model the classifier accuracy. This paper, on the other hand, attempts to make a formal connection between submodular functions, and accuracy functions for data subset selection and active learning. The closest work to this paradigm is the work of (Guillory & Bilmes, 2010; Golovin & Krause, 2010; Cuong et al., 2010) which models version space reduction via adaptive submodular functions. While this focuses on the fully adaptive setting, (Chen & Krause, 2013) extend this, to what we call the mini-batch adaptive setting. In this paper, we show that the data likelihood functions in the supervised setting are closely related to submodularity. Moreover, we show how we can extend this to a mini-batch adaptive active learning setting.

A number of papers on the other hand, do not use submodularity for data selection. For example, another common approach for training data summarization in the supervised and unsupervised setting is to use the concept of a coreset (Agarwal et al., 2005). The paradigm of coresets aims to efficiently approximate various geometric extent measures over a large set of data instances via a small subset. Many machine learning problems including SVMs, k-means clustering, k-median clustering, and Gaussian mixtures (Tsang et al., 2005; Har-Peled & Mazumdar, 2004; Feldman et al., 2011) can be approximately solved on a coreset.

Other algorithms having an active learning flavor include selecting the most informative set of items at every round (Settles, 2010), where the *informativeness* refers to utility of

the items from the classifiers' point of view. We measure the informativeness, by the classifier's uncertainty (in the case of *uncertainty sampling*) (Lewis & Gale, 1994), or the variance (in the case of *Query by Committee*) (Seung et al., 1992). It is interesting to note that some of these methods, e.g., uncertainty sampling, are special cases of adaptive submodular maximization (Cuong et al., 2010). These techniques do not ideally capture the *representativeness* of the samples, a problem further aggravated in the mini-batch adaptive setting. By *representativeness*, we mean how well a set of items covers the entire training set. This aspect is naturally modeled by density based methods (Nguyen & Smeulders, 2004). In order to obtain the benefits of both classes of algorithms, several papers have combined both notions (informativeness and representativeness) in a single objective (Xu et al., 2003; Huang et al., 2010). In this paper, we show that we can naturally incorporate both these notions in our multistage active learning algorithm FASS.

2 Naïve Bayes Classifier

We first consider the class of Naïve Bayes classification problems. Let $V = \{(x^i, y^i)\}_{i=1}^m$ be a set of training samples, where $x^i \in \mathcal{X}^d$ is a d-dimensional feature vector, and each feature takes a value from the finite set $\mathcal{X}$; each sample's label $y^i \in \mathcal{Y}$ takes a value from the finite set $\mathcal{Y}$ of classes. The ground set V may be partitioned as $V = V^1 \cup V^2 \cup \cdots \cup V^{|\mathcal{Y}|}$, where V^y is the set of all samples in V with class label y. We write the jth dimension of a feature vector x as $x_j \in \mathcal{X}$. We denote the maximum likelihood (ML) estimate of the parameters θ of a Naïve Bayes model, given a set of training samples $S \subseteq V$, as $\theta(S)$, and also use $\theta_{x_j|y} = p(x_j|y)$ and $\theta_y = p(y)$. For simplicity, we first assume no smoothing occurs during estimation but this will be considered later. The ML parameter function $\theta(S)$ can be given as follows:

$$\theta_{x_j|y}(S) = \frac{m_{x_j,y}(S)}{m_y(S)}; \tag{1}$$

and

$$\theta_y(S) = \frac{m_y(S)}{|S|} \tag{2}$$

where $m_{x_j,y}(S) = \sum_{i \in S} 1\{x_j^i = x_j \wedge y^i = y\}$ and $m_y(S) = \sum_{i \in S} 1\{y^i = y\}$. By definition, $m_{x_j,y}(S)$ counts the number of samples in S whose class label is $y \in \mathcal{Y}$ and whose jth dimension feature takes value $x_j \in \mathcal{X}$. Similarly, $m_y(S)$ counts the number of samples in S whose class label is $y \in \mathcal{Y}$. Both $m_{x_j,y}(S)$ and $m_y(S)$ are modular set functions, i.e., for any $S \subseteq V$, $m_{x_j,y}(S) = \sum_{s \in S} m_{x_j,y}(s)$ and $m_y(S) = \sum_{s \in S} m_y(s)$.

Given the parameter function $\theta(S)$, we introduce the notion of *data log-likelihood set function* $\ell^{\text{NB}} : 2^V \to \mathbb{R}$ that maps from each set $S \subseteq V$ of training samples to a log likelihood evaluated on the whole data set V:

$$\ell^{\text{NB}}(S) = \sum_{i \in V} \log p(x^i, y^i; \theta(S)), \tag{3}$$

$$\begin{aligned}
\ell^{\text{NB}}(S) &= \sum_{i \in V} \log p(x^i|y^i; \theta(S)) + \log p(y^i; \theta(S)) \\
&= \sum_{i \in V} \sum_{j=1}^{d} \log p(x_j^i|y^i; \theta(S)) + \sum_{i \in V} \log p(y^i; \theta(S)) \\
&= \sum_{i \in V} \sum_{j=1}^{d} \log \theta_{x_j^i|y^i}(S) + \sum_{i \in V} \log \theta_{y^i}(S) \\
&= \sum_{i \in V} \sum_{j=1}^{d} \log \frac{m_{x_j^i,y^i}(S)}{m_{y^i}(S)} + \sum_{i \in V} \log \frac{m_{y^i}(S)}{|S|} \\
&= \underbrace{\sum_{j=1}^{d} \sum_{x_j \in \mathcal{X}} \sum_{y \in \mathcal{Y}} m_{x_j,y}(V) \log(m_{x_j,y}(S))}_{\text{term 1: } f_{\text{NB}}(S)} \\
&\quad \underbrace{- (d-1) \sum_{y \in \mathcal{Y}} m_y(V) \log(m_y(S))}_{\text{term 2}} \underbrace{- |V| \log |S|}_{\text{term 3}}.
\end{aligned}$$

Figure 1. The NB likelihood as a function of S.

where $p(x^i, y^i; \theta(S))$ is the likelihood of the sample i parameterized by $\theta(S)$. $\ell^{\text{NB}}(S)$ acts as a utility set function for training a NB classifier. Under the Naïve Bayes assumption, we can express $\ell^{\text{NB}}(S)$ as shown in Figure 1.

Since $m_{x_j,y}(V)$, $m_y(V)$, and $|V|$ are all independent of S, they can be treated as constants in $\ell^{\text{NB}}(S)$. Then the first, second, and third terms of $\ell^{\text{NB}}(S)$ are in the form of a sum of concave over modular functions, hence they are all monotone submodular (Ahmed & Atamtürk, 2009; Stobbe & Krause, 2010; Lin & Bilmes, 2011). As a result, $\ell^{\text{NB}}(S)$ is in the form of difference of submodular functions, and the underlying optimization problem becomes a difference of submodular (DS) optimization:

$$\max_{|S|=k} \ell^{\text{NB}}(S). \tag{4}$$

While there are scalable heuristics that work well in practice to minimize a difference of submodular functions (Narasimhan & Bilmes, 2005; Iyer & Bilmes, 2012; Iyer et al., 2014), these techniques lack worst-case guarantees since the underlying problem is hard to optimize.

Fortunately, the second and the third term of $\ell^{\text{NB}}(S)$ become constants when we enforce a set of inconsequential constraints. We call the first term (the only active term in a transformed optimization problem) the *Naïve Bayes submodular function*:

$$f_{\text{NB}}(S) = \sum_{j=1}^{d} \sum_{x_j \in \mathcal{X}} \sum_{y \in \mathcal{Y}} m_{x_j,y}(V) \log m_{x_j,y}(S). \tag{5}$$

Given the equality constraint $|S| = k$, the third term in $\ell^{\text{NB}}(S)$ becomes a constant in Problem 4. Furthermore,

we make an assumption that the selected set should be balanced, which makes the second term also a constant. In particular, we say that a set S is *balanced* if S maintains the same distribution over the class labels as the whole data set. A set S of size k is balanced if the proportion of each class label in the set S is the same as the whole data set V, i.e., $|S \cap V^y| = k\frac{|V^y|}{|V|}$ for any $y \in \mathcal{Y}$ for some k. We assume that k is chosen such that $k\frac{|V^y|}{|V|}$ is an integer for all $y \in \mathcal{Y}$, and if not then we round $k\frac{|V^y|}{|V|}$ to the closest integer. With balance enforced and $|S| = k$, we have that $m_y(S) = |S \cap V^y| = k\frac{|V^y|}{|V|}$ for all $y \in \mathcal{Y}$. Therefore, term 2 of $\ell^{\text{NB}}(S)$ also becomes a constant. Let $\mathcal{M}(V, I)$ be a partition matroid using the partition $\{V^y\}_{y\in\mathcal{Y}}$, where the set of bases $\mathcal{B}(\mathcal{M})$ is defined as $\mathcal{B}(\mathcal{M}) = \{S \subseteq V : |S \cap V^y| = k\frac{|V^y|}{|V|}, \forall y \in \mathcal{Y}\}$. Thus, a set S of size k being balanced is equivalent to S being a base of the matroid $\mathcal{M}$, i.e., $S \in \mathcal{B}(\mathcal{M})$. Since the second and the third term of $\ell^{\text{NB}}(S)$ are constants given the constraint $S \in \mathcal{B}(\mathcal{M})$, the above optimization problem is equivalent to:

$$\max_{S\in\mathcal{B}(\mathcal{M})} f_{\text{NB}}(S). \tag{6}$$

We may efficiently, scalably, and approximately solve this problem using the lazy greedy algorithm (Fisher et al., 1978). This formulation asks for a small training data subset S such that the likelihood of the ML parameters $\theta(S)$ is large on the *entire* data set V, and the following Theorem offers perspective in terms of the KL-divergence.

Theorem 1. *Let* $D_{KL}(p(x, y; \theta(V))||p(x, y; \theta(S))) \triangleq \sum_{x\in\mathcal{X}^d}\sum_{y\in\mathcal{Y}} p(x, y; \theta(V)) \log \frac{p(x,y;\theta(V))}{p(x,y;\theta(S))}$ *be the KL-divergence between* $p(x, y; \theta(V))$ *and* $p(x, y; \theta(S))$*, where* $p(x, y; \theta(S))$ *is the maximum likelihood estimate of the joint distribution given a data set* S*. Under the Naïve Bayes assumption, Problem 4 is equivalent to the following:*

$$\min_{|S|=k} D_{KL}(p(x, y; \theta(V))||p(x, y; \theta(S))). \tag{7}$$

Proof. We derive D_{KL} as in Figure 2. Then we have

$$D_{KL}(p(x, y; \theta(V))||p(x, y; \theta(S))) = -\frac{1}{|V|}\ell^{\text{NB}}(S) + C, \tag{8}$$

therefore, Problem 7 is equivalent Problem 4 □

We next point out connections between the Naïve Bayes submodular function f_{NB} and the class of feature-based submodular functions f_{fea} defined in (Wei et al., 2014b) and that have effectively been applied in various subset selection tasks (Shinohara, 2014; Tschiatschek et al., 2014; Wei et al., 2014c;b; Kirchhoff & Bilmes, 2014). These are defined as $f_{\text{fea}}(S) = \sum_{u\in\mathcal{U}} w_u g(c_u(S))$, where g is a concave function, $\mathcal{U}$ is a set of "features", $\{w_u\}_{u\in\mathcal{U}}$ is a set of non-negative weights for each feature $u \in \mathcal{U}$, and $c_u(S) = \sum_{s\in S} c_u(s)$ is a non-negative modular score for feature

$$\begin{aligned}
&D_{KL}(p(x, y; \theta(V))||p(x, y; \theta(S))) \\
&= -\sum_{x_1,\ldots,x_d\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x_1, \ldots, x_d, y; \theta(V)) \log p(x_1, \ldots, x_d, y; \theta(S)) \\
&+ \underbrace{\sum_{x_1,\ldots,x_d\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x_1, \ldots, x_d, y; \theta(V)) \log p(x_1, \ldots, x_d, y; \theta(V))}_{\text{constant: } C} \\
&= -\sum_{x_1,\ldots,x_d\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x_1, \ldots, x_d, y; \theta(V)) \sum_{j=1}^{d} \log p(x_j|y; \theta(S)) \\
&- \sum_{x_1,\ldots,x_d\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x_1, \ldots, x_d, y; \theta(V)) \log p(y; \theta(S)) + C \\
&= -\sum_{j=1}^{d}\sum_{x_j\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x_j, y; \theta(V)) \log p(x_j|y; \theta(S)) \\
&- \sum_{y\in\mathcal{Y}} p(y; \theta(V)) \log p(y; \theta(S)) + C \\
&= -\sum_{j=1}^{d}\sum_{x_j\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \frac{m_{x,y}(V)}{|V|} \log \frac{m_{x,y}(S)}{m_y(S)} - \sum_{y\in\mathcal{Y}} \frac{m_y(V)}{|V|} \log \frac{m_y(S)}{|S|} + C \\
&= \frac{1}{|V|}\{-\sum_{j=1}^{d}\sum_{x_j\in\mathcal{X}}\sum_{y\in\mathcal{Y}} m_{x_j,y}(V) \log m_{x_j,y}(S) + |V| \log |S| \\
&+ (d-1)\sum_{y\in\mathcal{Y}} m_y(V) \log m_y(S)\} + C \\
&= -\frac{1}{|V|}\ell^{\text{NB}}(S) + C
\end{aligned}$$

Figure 2. Derivation of $D_{KL}(p(x, y; \theta(V))||p(x, y; \theta(S)))$.

$u \in \mathcal{U}$ in S, with $c_u(s)$ measuring the degree to which item s "possesses" feature u. Defining $\mathcal{U}$ as the set of all possible input-label pairs, i.e., $\mathcal{U} = \{(x_j, y)|x_j \in \mathcal{X}, y \in \mathcal{Y}, j = 1, \ldots, d\}$, the weight for each feature as $w_u = m_u(V)$, the concave function as $g(x) = \log x$, and the modular score as $c_u(S) = m_u(S)$, f_{NB} is then an instance of a feature-based function. Maximizing f_{NB} chooses data that has diverse coverage in the set of features $\mathcal{U}$ where the desired coverage for each feature $u \in \mathcal{U}$ is controlled by its weight $w_u = m_u(V)$.

Laplace smoothing: Without any smoothing, it may be that the Naïve Bayes submodular function f_{NB} is undefined at $S = \emptyset$. Smoothing not only fixes this issue, it is naturally incorporated into the submodular framework. Given a Laplace smoothing parameter $\alpha > 0$, for a set $S \subseteq V$, the Laplace smoothed ML estimated parameters become

$$\theta^{\alpha}_{x_j|y}(S) = \frac{m_{x_j,y}(S) + \alpha}{m_y(S) + \alpha|\mathcal{X}|}; \tag{9}$$

$$\theta_y^{\alpha}(S) = \frac{m_y(S) + \alpha}{|S| + \alpha|\mathcal{Y}|}. \tag{10}$$

We may formulate this similar to Problem 6, where the objective is slightly modified from before. First, define a slightly expanded ground set $V' = V \cup \{v'\}$ where v' is a pseudo-sample that has the property $m_{x_j,y}(v') = \alpha, \forall x_j \in \mathcal{X}, y \in \mathcal{Y}, j = 1, \ldots, d$. We next define a function $f'_{\text{NB}(\alpha)} : 2^{V'} \to \mathbb{R}_+$ as:

$$f'_{\text{NB}(\alpha)}(S) = \sum_{j=1}^{d} \sum_{x_j \in \mathcal{X}} \sum_{y \in \mathcal{Y}} m_{x_j,y}(V) \log(m_{x_j,y}(S)) \tag{11}$$

For any $S \subseteq V$, $f'_{\text{NB}(\alpha)}(S \cup \{v'\})$ represents the Laplace-smoothed ML objective. From this, we can obtain a normalized monotone non-decreasing submodular function $f_{\text{NB}(\alpha)} : 2^V \to \mathbb{R}_+$ where $f_{\text{NB}(\alpha)}(S) = f'_{\text{NB}(\alpha)}(S \cup \{v'\}) - f'_{\text{NB}(\alpha)}(\{v'\})$ whose score is the Laplace-smoothed ML estimate minus a constant.

Naïve Bayes in Text Classification: Next, we consider the problem of text classification, where Naïve Bayes classifier under the bag-of-words model often performs very well. In this context, $V = \{(D^i, y^i)\}_{i \in V}$ is a set of document-label pairs, $\mathcal{Y}$ is a set of document labels (e.g., topics), each document D^i is assigned to only one label $y^i \in \mathcal{Y}$. We represent each document D^i as a bag of words $D^i = \{w_j\}_{j=1}^{n_i}$, where n_i is the number of words in the document, and each word w_j is taken from a vocabulary $\mathcal{W}$. Let $c_y(S) = \sum_{i \in S} 1\{y^i = y\} n_i$ counts the number of words in the set S of documents labeled as y. $c_y(S)$ is a modular function for each label $y \in \mathcal{Y}$. We also define $m_{w,y}(S) = \sum_{i \in S} m_{w,y}(i)$ where $m_{w,y}(i) = \sum_{w' \in D_i} 1\{w' = w \wedge y^i = y\}$. By definition, $m_{w,y}(S)$ counts the number of occurrences of $w \in \mathcal{W}$ in the subset S of documents that are labeled as y. Following the same spirit, we define the data log likelihood function $\ell^{\text{NB}}_{\text{text}}(S)$ with the following form:

$$\ell^{\text{NB}}_{\text{text}}(S) = \sum_{i \in V} \log p(D^i, y^i; \theta(S)). \tag{12}$$

Under the Naïve Bayes assumption, we can simplify it as follows:

$$\begin{aligned}
\ell^{\text{NB}}_{\text{text}}(S) &= \sum_{i \in V} \log p(D^i | y^i; \theta(S)) + \sum_{i \in V} \log p(y^i; \theta(S)) \\
&= \sum_{i \in V} \sum_{w \in D^i} \log p(w | y^i; \theta(S)) + \sum_{i \in V} \log p(y^i; \theta(S)) \\
&= \sum_{i \in V} \sum_{w \in D^i} \log \frac{m_{w,y^i}(S)}{c_{y^i}(S)} + \sum_{i \in V} \log \frac{m_{y^i}(S)}{|S|} \\
&= \underbrace{\sum_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} m_{w,y}(V) \log m_{w,y}(S)}_{\text{term 1: } f_{\text{NB-text}}} - \underbrace{|V| \log |S|}_{\text{term 2}} \\
&\quad - \underbrace{\sum_{y \in \mathcal{Y}} c_y(V) \log c_y(S)}_{\text{term 3}} + \underbrace{\sum_{y \in \mathcal{Y}} m_y(V) \log m_y(S)}_{\text{term 4}}
\end{aligned}$$

The data log likelihood function $\ell^{\text{NB}}_{\text{text}}(S)$ is again in the form of difference of submodular functions. By enforcing the chosen set S to be of fixed size and balanced, term 2 and 4 can be handled as constants. Furthermore, term 3 can also be a constant if each document has the same length d, i.e., $n_i = d$ for all $i \in V$. This can be a reasonable assumption to make, since one can always normalize the word counts for each document such that each document has constant length with potentially fractional word counts, and more importantly, it has been reported in (Nigam et al., 2000) that better bag-of-words NB model may be trained if the training documents are word-count normalized. We formulate the supervised data subset selection for text classification problem under the bag-of-words Naïve Bayes model as follows:

$$\max_{S \in \mathcal{B}(\mathcal{M})} \ell^{\text{NB}}_{\text{text}}(S), \tag{13}$$

which can be equivalently transformed to the following:

$$\max_{S \in \mathcal{B}(\mathcal{M})} f_{\text{NB-text}}(S), \tag{14}$$

where $f_{\text{NB-text}}(S) = \sum_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} m_{w,y}(V) \log m_{w,y}(S)$. The same Laplace smoothing technique can also be naturally incorporated into this framework and make $f_{\text{NB-text}}$ well-defined at $S = \emptyset$.

3 Nearest Neighbor Classification

In this section, we consider (non-parametric) Nearest Neighbor (NN) classifiers and formulate supervised data subset selection problem as constrained submodular maximization. Given a set of training samples $V = \{(x^i, y^i)\}_{i=1}^m \subseteq \mathcal{X} \times \mathcal{Y}$ and a similarity function $w : V \times V \to \mathbb{R}^+$ which measures the similarity between any pair of data instances in feature space, the NN classifier simply classifies $x \in \mathcal{X}$ based on its closest training sample. The similarity between training sample pairs i and j is given as $w(i,j) = d - \|x^i - x^j\|_2^2 \geq 0$, where $d = \max_{v \in V, v' \in V} \|x^v - x^{v'}\|_2^2$ is the maximum pairwise distance. Though extremely simple, the NN classifier has been applied on a number of machine learning tasks, including hand-written digit recognition, text categorization, object recognition, etc (Bhatia et al., 2010; Boiman et al., 2008; Shah et al., 2011). For a NN classifier, no model needs to be "learnt" as nearly all the computation takes place at the classification stage. The complexity of classifying a sample can be expensive and is dependent on the number of training samples. A way to alleviate this problem is to reduce the training set size ideally without losing performance, a problem well suited to supervised data subset selection.

Similar to the NB classifier analysis, we consider a data log-likelihood set function $\ell^{\text{NN}} : 2^V \to \mathbb{R}$ that maps each subset $S \subseteq V$ to a log likelihood score on the whole set V:

$$\ell^{\text{NN}}(S) = \sum_{i \in V} \log p(x^i | y^i; \theta(S)) + \sum_{i \in V} \log p(y^i; \theta(S)),$$

where $p(x^i | y^i; \theta(S))$ and $p(y^i; \theta(S))$ are the generative likelihood and the prior likelihood of the sample $i \in V$ given

by $\theta(S)$. The idea of the data subset selection is to select a small sized set S so that $\ell^{\text{NN}}(S)$ is maximized. As in the NB scenario, we express the prior as $p(y^i;\theta(S)) = \frac{m_{y^i}(S)}{|S|}$. The key question regarding the function ℓ^{NN} is how to appropriately characterize the generative likelihood function $p(x^i|y^i;\theta(S))$ so that it is of a simple form leading ℓ^{NN} to be submodular and also maps well to the NN classifier. To this end, we assume the following:

1. $p(x^i|y^i;\theta(S))$ is determined only by the sample j in S that is with label y^i and is closest to i, i.e., $j \in \text{argmax}_{s\in S\cap V^{y^i}} w(i,s)$;
2. The generative likelihood is expressed as $p(x^i|y^i;\theta(S)) = ce^{-\|x^i-x^j\|_2^2} = ce^{w(i,j)-d} = c'e^{w(i,j)} = c'\exp\left(\max_{s\in S\cap V^{y^i}} w(i,s)\right)$, where c and c' are constants.

We express the generative log-likelihood as $\log p(x^i|y^i;\theta(S)) = \log c' + \max_{s\in S\cap V^{y^i}} w(i,s)$ yielding:

$$\ell^{\text{NN}}(S) = \underbrace{\sum_{y\in\mathcal{Y}}\sum_{i\in V^y}\max_{s\in S\cap V^y} w(i,s)}_{\text{term 1: } f_{\text{NN}}} + \underbrace{\sum_{y\in\mathcal{Y}} m_y(V)\log m_y(S)}_{\text{term 2}} - \underbrace{|V|\log|S|}_{\text{term 3}} + \underbrace{C}_{\text{constant}}.$$

The first term is the *Nearest Neighbor submodular function*:

$$f_{\text{NN}}(S) = \sum_{y\in\mathcal{Y}}\sum_{i\in V^y}\max_{s\in S\cap V^y} w(i,s). \tag{15}$$

Similar to the NB case, $\ell^{\text{NN}}(S)$ is a difference of submodular functions. In a manner similar to the NB classifier, we assume that the selected set is balanced and of fixed size k. The second and the third term of ℓ^{NN} are treated as constants. Hence, the problem

$$\max_{S:|S|=k} \ell^{\text{NN}}(S) \tag{16}$$

is equivalently expressed as constrained submodular maximization:

$$\max_{S\in\mathcal{B}(\mathcal{M})} f_{\text{NN}}(S). \tag{17}$$

Connection to facility location function: We next show f_{NN}'s connections to the *facility location* function (Mirchandani & Francis, 1990), defined as :

$$f_{\text{fac}}(S) = \sum_{i\in V}\max_{j\in S} w(i,j). \tag{18}$$

Facility location function is often applied to identify representative instances from a big collection of items (Zheng et al., 2014; Wei et al., 2013; Lin & Bilmes, 2011; Gomes & Krause, 2010; Iyer & Bilmes, 2013). Sharing very similar definitions, the facility location function f_{fac} is in fact a special case of f_{NN}, when all items in V take the same class labels, or equivalently, $|\mathcal{Y}| = 1$. Given its resemblance to f_{NN}, the facility location function should naturally model the utility of data sets for training NN classifiers, although it was originally designed to measure the representativeness of each set S about the whole set V. Also, f_{NN} can be written as a sum of facility location functions since $f_{\text{NN}}(S) = \sum_{y\in\mathcal{Y}} f^y_{\text{fac}}(S\cap V^y)$ with $f^y_{\text{fac}}(S) = \sum_{i\in V^y}\max_{j\in S} w(i,j)$ a facility location with ground set V^y. As far as we know, this is the first work to 1) connect the facility location function to the utility of training NN classifiers, and 2) to show that the utility of a set for training NN classifiers is measured by its representativeness about the data partition for each class.

Scalability of f_{NN}: To instantiate f_{NN}, one needs to compute the similarity for every pair of data instances in each block V^y. Equivalently, we can define f_{NN} via a similarity graph $G_{\text{NN}}(V,E)$, where each block V^y constitutes a complete graph and every pair of vertices $v, v' \in V^y$ is connected with edge weight $w(v,v')$. Similarly, the facility location function f_{fac} is defined via a complete similarity graph $G_{\text{fac}}(V,E)$. Therefore, we call both functions *graph-based submodular* functions. The time and memory complexity for constructing and storing the similarity graph G_{NN} and G_{fac} is, in the worst case, $O(|V|^2)$. The applicability of f_{NN} and f_{fac} is thus significantly limited when $|V|$ is large. (Wei et al., 2014a) addresses this problem for f_{fac} by approximating f_{fac} with a surrogate function f'_{fac} that is defined on a K-Nearest Neighbor sparse sub-graph of G_{fac}. In this work, we use the same idea and utilize a surrogate function f'_{NN} for f_{NN}. Instead of being instantiated on G_{NN}, we define the surrogate function f'_{NN} on its K-Nearest Neighbor sparse sub-graph $\hat{G}^K_{\text{NN}}$, where each vertex $v \in V$ of label y is connected only to its K most similar neighbors in the block V^y. As a result, significant reduction on the memory and time complexity of evaluating f_{NN} is achieved. Define $\hat{w}$ as the edge weights on the K-Nearest Neighbor sparse sub-graph $\hat{G}^K_{\text{NN}}$. Then we have $\hat{w}(i,j) = w(i,j)$ if the item j of label y^j is among the K nearest neighbor of i in the partition V^{y^j}, and $\hat{w}(i,j) = 0$, otherwise. To establish that the surrogate function f'_{NN}, even with very sparse K, is a good approximation of f_{NN}, we rely on a key observation: $\max_{j\in S\cap V^{y^i}} w(i,j) = \max_{j\in S\cap V^{y^i}} \hat{w}(i,j)$ holds if the set S contains at least one item that is among the K nearest neighbor of i in the partition V^{y^i}. Thus, showing that $f'_{\text{NN}}(S) = f_{\text{NN}}(S)$ is equivalent as showing that the set S contains at least one item that is among the K nearest neighbor of item i in the partition V^{y^i} for any $i \in V$. We denote that $|\mathcal{Y}| = C$. For simplicity, we assume that each block of the data set is balanced, i.e., $|V^y| = d$ for any $y \in \mathcal{Y}$, and then the ground set size $n = |V| = Cd$. Given any data item i with label y, let's denote $\vec{w}_i = \{w(i,1),\ldots,w(i,d)\}$ as the vector containing the weights on all edges connecting vertex i to other items in the same partition V^y. To this end, we assume that the ranking of any item j among

the vector $\vec{w}_i$ for any $i \in V$ is uniformly distributed over $\{1, 2, \ldots, d\}$ and that the ranking of j in one weight vector $\vec{w}_i$ is independent of its ranking in another.

Lemma 1. *For the Nearest Neighbor submodular function, we have:*

$$f_{NN}(S) = f'_{NN}(S), \forall S \subseteq V \text{ s.t. } |S| \geq \alpha n, \quad (19)$$

with probability at least $(1-\theta)$, and the sparsity of the K Nearest Neighbor sparse graph $\hat{G}^K_{NN}$ being at least

$$K = d[1 - (\frac{\theta}{n})^{\frac{C}{\alpha n}}]. \quad (20)$$

Proof. Let $\vec{\hat{w}}_i$ be the ith row vector obtained from the K Nearest Neighbor sparse approximation of the full graph. Then, $\vec{\hat{w}}_i$ is the approximate vector for $\vec{w}_i$ with only K largest values retained. The key observation for the Nearest Neighbor submodular function is that $f_{\text{NN}}(S) = f'_{\text{NN}}(S)$, if the set S contains items that are among the top K values of the row vector $\vec{w}_i$ for all i, since $\max_{j \in S \cap V^{y^i}} w(i,j) = \max_{j \in S \cap V^{y^i}} \hat{w}(i,j)$, if S contains items that are among the top K values of $\vec{w}_i$.

For notation simplicity, we write $\hat{f}$ for f'_{NN} and f for f_{NN}. For any item $t \in V$, we have the probability of $w(i,t)$ not being among the top K elements of the row vector w_i as $\frac{d-K}{d}$, given the uniform distribution assumption.

We denote $m = \alpha n$. By the independence assumption, the probability, for which a set S_m of size m contain at least one item among the top K elements for each row vector, can be then computed as $[1 - (\frac{d-K}{d})^{\frac{m}{C}}]^n$.

Let the probability that S_m covers among the top K elements of all row vectors be $1-\theta$. Then, we have the following:

$$[1 - (\frac{d-K}{d})^{m/C}]^n = 1 - \theta$$

Simplifying the equation, we can get the following:

$$K = d[1 - (1 - (1-\theta)^{\frac{1}{n}})^{\frac{C}{m}}] \quad (21)$$

$$\approx d[1 - (1 - e^{-\frac{\theta}{n}})^{\frac{C}{m}}] \quad (22)$$

$$\approx d[1 - (\frac{\theta}{n})^{\frac{C}{m}}] \quad (23)$$

The first approximation follows since $(1-\theta)^{\frac{1}{n}} \approx e^{-\frac{\theta}{n}}$, for θ being close to 0. The second approximation follows from that $e^{-\frac{\theta}{n}} \approx 1 - \frac{\theta}{n}$, with $-\frac{\theta}{n} \approx 0$. □

Assuming that θ is a constant, we have $\lim_{n\to\infty} d[1 - (\frac{\theta}{n})^{\frac{C}{\alpha n}}] = \frac{1}{\alpha}\log n + \text{constant}$. The Lemma implies that with high probability $1-\theta$, f'_{NN} and f_{NN} share the same function value for any sets of size greater than some threshold αn, where $\hat{G}^K_{\text{NN}}$ can be as sparse as $K = O(\log n)$.

Extension to k-Nearest Neighbor classifiers: Next, we extend the analysis of the likelihood function to the k-Nearest Neighbor classifiers for $k > 1$. Consider a data log-likelihood set function $\ell^{\text{kNN}} : 2^V \to \mathbb{R}$ that maps from each subset $S \subseteq V$ to a log-likelihood score of the k-NN classifier on the whole data set V:

$$\ell^{\text{kNN}} = \sum_{i \in V} \log p(x^i | y^i; \theta(S)) + \sum_{i \in V} \log p(y^i; \theta(S)). \quad (24)$$

It is straightforward to write the prior likelihood as $p(y^i; \theta(S)) = \frac{m_{y^i}(S)}{|S|}$. The remaining question is how to characterize the generative likelihood function $p(x^i | y^i; \theta(S))$ given a set of items $S \subseteq V$. In the context of a k-NN classifier, we assume the followings:

1. The similarity $w(i,j)$ between any pair of items i and j is computed as $w(i,j) = d - \|x^i - x^j\|_2^2$, where $d = \max_{i \in V, j \in V} \|x^i - x^j\|_2^2$ is the maximum pairwise Euclidean distance.
2. $p(x^i | y^i; \theta(S))$ is determined by the set of k samples $T \subseteq S$ that are with label y^i and are closest to i, i.e., $T \in \arg\max_{|T|=k; T \subseteq S \cap V^{y^i}} \sum_{t \in T} w(t,i)$, (here we assume $|S \cap V^{y^i}| \geq k$)
3. The generative likelihood is parameterized as

$$p(x^i | y^i; \theta(S)) \propto \prod_{t \in T} e^{-\|x^i - x^t\|_2^2} \quad (25)$$

$$= c \prod_{t \in T} e^{w_{i,t} - d} \quad (26)$$

$$= c' \exp\{\max_{T \subseteq S \cap V^{y^i}; |T|=k} \sum_{t \in T} w(i,t)\} \quad (27)$$

Using this, we derive the log generative likelihood as $\log p(x^i | y^i; \theta(S)) = \log c' + \max_{T \subseteq S \cap V^{y^i}; |T|=k} \sum_{t \in T} w(i,t)$ leading to the following:

$$\ell^{\text{kNN}}(S) = \underbrace{\sum_{y \in \mathcal{Y}} \sum_{i \in V^y} \max_{T \subseteq S \cap V^y; |T|=k} \sum_{t \in T} w(i,t)}_{\text{term 1: } f_{\text{k-NN}}}$$
$$+ \underbrace{\sum_{y \in \mathcal{Y}} m_y(V) \log m_y(S)}_{\text{term 2}} - \underbrace{|V| \log |S|}_{\text{term 3}} + \underbrace{C}_{\text{constant}}.$$

Given the same assumption about the chosen set to be balanced, the first term is the only effective term in the transformed optimization problem. We call the first term *k-Nearest Neighbor submodular function* with the form $f_{\text{kNN}}(S) = \sum_{y \in \mathcal{Y}} \sum_{i \in V^y} \max_{T \subseteq S \cap V^y; |T|=k} \sum_{t \in T} w(i,t)$, which interestingly turns out to be in form of the weighted matroid rank functions as defined in (Shioura, 2009).

4 Active Learning

We next extend the results of supervised data subset selection to the active learning setting, where we incrementally obtain labels. We define a multistage *mini-batch adaptive active learning* framework, where selection adapts to the labels previously obtained. Moreover, we show how we can naturally combine the notions of *representativeness* and *information* by filtering. In this section, we focus on uncertainty sampling to represent *information*, but our methods can extend to other strategies, such as Query by Committee, as well.

In the mini-batch active learning setting, the algorithm iteratively selects a set of B unlabeled instances to label at every round, and this is done for T rounds. Later rounds get to use the labels previously selected, so this is an adaptive strategy, but within each mini-batch all labels are selected simultaneously without mutual knowledge of or interaction with each other. In the end, we obtain $k = BT$ labeled instances. We denote the set of unlabeled instances as $\mathcal{U}$, and the goal is to select a labeled set $\mathcal{L}$ such that $|\mathcal{L}| = k$. A common strategy is *uncertainty sampling*, where the B most uncertain examples (from the current classifiers perspective) are chosen for labeling (Lewis & Gale, 1994) at every round. Given a round t and a set of labeled items $\mathcal{L}$, let $\delta_u^t \geq 0$ be the uncertainty score for an example $u \in \mathcal{U} \setminus \mathcal{L}$. The uncertainty sampling approach, then, simply selects $S \in \max_{S' \subseteq \mathcal{U} \setminus \mathcal{L}; |S'|=B} \sum_{u \in S'} \delta_u^t$, and adds these to the labeled set $\mathcal{L}$. The drawback of this approach is that it fails to model the interactions between samples, i.e., labeling one sample could often affect the utility of labeling another. Simply choosing the most uncertain samples might lead to a selected set with high redundancy. A better strategy would choose a diverse set of samples from amongst those that the currently trained model is most uncertain about.

We propose a multi-stage batch active learning scheme called *filtered active submodular selection (*FASS*)*. This algorithm (see Alg. 1) attempts to solve the original data subset selection problem of maximizing a submodular function f (i.e., either the Naïve Bayes submodular function f_{NB}, or the NN submodular function f_{NN}) in an iterative manner. At every round t, we first filter out data samples that the current model is certain about, and preserve a candidate set of β_t ($\beta_t \geq B$) most uncertain samples. Specifically, we find a solution to $\mathcal{U}^t \in \operatorname{argmax}_{U' \subseteq \mathcal{U} \setminus \mathcal{L}; |U'|=\beta_t} \sum_{u \in U'} \delta_u^t$. Since we do not know the labels of the items in $\mathcal{U}^t$, we use the most probable prediction (based on the current classifier) $\hat{y}_u$ for each item $u \in \mathcal{U}^t$ as its hypothesized label. We then instantiate an appropriate submodular objective $\hat{f}_t : 2^{\mathcal{U}^t} \to \mathbb{R}_+$, which has essentially the same form as f, except that it is defined on the ground set $\mathcal{U}^t$, and uses the hypothesized labels $\hat{y}_u, \forall u \in \mathcal{U}^t$. We then solve the optimization problem

$$\max_{|S|=B; S \subseteq \mathcal{U}^t} \hat{f}_t(S). \tag{28}$$

The scheme of FASS is fully characterized by the choice of the monotone submodular objective f, the scaling parameters $\{\beta_t\}_{t=1}^T$ and the classifier. Given a classifier, better performance of FASS is expected when the submodular objective f matches the utility function for training the classifier. Hence in the case of the NB classifier, we use f_{NB} as the submodular function, while in the case of NN classifier, we can use f_{NN} or f_{fac}. In general however, we can use any submodular function f in our framework. In Section 5, we show that FASS with f_{NB} yields superior performance in the case of NB classifiers, while FASS with f_{fac} or f_{NN} performs better on NN classifiers.

Algorithm 1 Filtered Active Submodular Selection

1: **Input:** $\mathcal{U}$, T, B, $\{\beta_t\}_{t=1}^T$, Starting set of labels $\mathcal{L}$
2: **for** $t = 1, \cdots, T$ **do**
3: Train the classifier using the labeled set $\mathcal{L}$, and derive the uncertainty scores δ^t;
4: $\mathcal{U}^t \in \operatorname{argmax}_{U \subseteq \mathcal{U} \setminus \mathcal{L}; |U|=\beta_t} \sum_{u \in U} \delta_u^t$;
5: Obtain the most probable labels as the hypothesized labels $\{\hat{y}_u\}_{u \in \mathcal{U}^t}$.
6: Instantiate $\hat{f}_t : 2^{\mathcal{U}^t} \to \mathbb{R}_+$ on the hypothesized labels $\{\hat{y}_u\}_{u \in \mathcal{U}^t}$ and ground set $\mathcal{U}^t$;
7: Find $L^t \in \operatorname{argmax}_{|S|=B; S \subseteq \mathcal{U}^t \setminus \mathcal{L}} \hat{f}_t(S)$.
8: $\mathcal{L} = \mathcal{L} \cup L^t$.
9: **end for**

Next, we discuss the scaling parameters $\{\beta_t\}_{t=1}^T$ for FASS. For any round t, β_t is the size of the candidate set $\mathcal{U}^t$ and controls the trade-off between the criteria of uncertainty and the submodular objective f. If $\beta_t = B, \forall t$, the selected set is chosen only accounting for the uncertainty scores, and FASS is reduced to the uncertainty sampling approach. When $\beta_t = |\mathcal{U} \setminus \mathcal{L}|$, the selected set does not account for the uncertainty scores, and is solely chosen by the submodular objective f. In our experiments, we set the scaling parameters at each round as constant β, i.e., $\beta_t = \beta, \forall t$. Choice of β affects the time and memory complexity of an instance of FASS scheme. It becomes significant when f is a graph-based submodular function, e.g., f_{NN} and f_{fac}. The time and memory complexity for constructing the similarity graph grows quadratically with β. Fortunately, we empirically observe that FASS often performs rather well for small values of β ($\beta \ll |\mathcal{U}|$). As a result, FASS can easily scale to extremely large data sets. Thanks to our uncertainty sampling based filtering and data selection via submodular maximization, we naturally incorporate notions both of *information* and *representativeness*. Moreover, thanks to the greedy algorithm for maximization, as well as the prefiltering we perform, our approach can easily scale to large real-world machine learning problems.

We also point out that FASS subsumes the submodular active learning framework in (Hoi et al., 2006) as it is a special case of FASS with $\beta_t = |\mathcal{U} \setminus \mathcal{L}|$ and f being chosen as the *Fisher information submodular function* f_{fs} defined as

$$f_{\text{fs}}(S) = \tfrac{1}{c} \sum_{i \in \mathcal{U}} \pi_i (1 - \pi_i) - \sum_{i \notin S} \frac{\pi_i (1 - \pi_i)}{c + \sum_{j \in S} \pi_i \pi_j (x_i^T x_j)^2},$$

where $c > 0$, π_i is the posterior probability of a sample i, and x_i is the feature representation for i. It is shown (Hoi

et al., 2006) that f_{fs} is normalized monotone submodular and approximates the utility of a given set S by how much it reduces the Fisher information for logistic regression classifier. We empirically evaluate f_{fs} in our experiments.

It is interesting to note that Algorithm 1 can be viewed as a multistage approximation framework for optimizing a data subset selection objective f. Moreover, the algorithm closely resembles the algorithm from (Wei et al., 2014a), and the functions $\hat{f}_t$ can be viewed as successive approximations of the original function f. An open problem is whether the results from (Wei et al., 2014a) can be extended to provide approximation guarantees for Algorithm 1.

5 Experimental results

We empirically evaluate the proposed framework on the supervised data subset selection problem and the active learning problem. In the set of experiments, we wish to address the following: 1) How Eqn 6 and 17 perform on the supervised data subset selection for NB and NN classifier, respectively; 2) How FASS performs on active learning under various choices of f and β; and 3) How well the proposed framework extend to other classifiers, including Logistic Regression (LR) and Deep Neural Networks (DNN). In our experiments, we evaluate on two separate tasks: 1) text categorization, where we tested three classifiers: NB, NN, and LR; and 2) handwritten digit recognition, where we evaluated NN and DNN classifiers.

5.1 Text Categorization Experiments

Experimental setup: We evaluate text categorization on the 20 Newsgroups data set [1], which consists of 18774 articles divided almost evenly among 20 different UseNet discussion groups (Lang, 1995). The goal is to classify an article into one newsgroup (of twenty) to which it was posted. It is a multi-class classification problem. For each instance of the experiment, we randomly split $\frac{2}{3}$ and $\frac{1}{3}$ of the whole data set as the training and test samples. Each subset selection strategy is applied to sub-select the training samples and then train a classifier. We report its classification error rate on the test set averaged over 20 instances of random data splits as the performance for each subset selection strategy.

For mini-batch active learning experiments, we first randomly label $B = 100$ samples, on which we train a classifier as the initial model. In each iteration, additional B unlabeled examples are selected for labeling to update the model. We evaluate for $T = 10$ iterations ending with a total of $k = 1000$ labeled examples. Under the least confident criterion, in each iteration t, we compute the uncertainty score δ_u^t of a sample u as $\delta_u^t = 1 - p(\hat{y}_u|x_u)$, where $\hat{y}_u$ is the most probable prediction of the sample u given by the currently trained model, i.e., $\hat{y}_u \in \operatorname{argmax}_{y\in\mathcal{Y}} p(y|x_u)$. We evaluate the proposed supervised data subset selection framework also on the same sequence of subset sizes so that it can be compared with the mini-batch active learning results. We construct a random sampling baseline for comparison, where we randomly sample the data set at appropriate sizes for labeling. The similarity between any pair of documents is defined as the cosine similarity between their TF-IDF representations. For FASS, we fix $\beta_t = \beta = 4000, \forall t$ and test four different submodular objectives: f_{NB}, f_{NN}, f_{fac}, and f_{fs} ($c = 0.1$). In addition, we construct another baseline (FASS+RS), which is implemented the same as FASS, except that in each iteration, the submodular optimization procedure in Line 7 is replaced with a random sub-sampling strategy.

Naïve Bayes Classifier: We first explore the NB classifier under the bag-of-words model. We apply a Laplace smoothing parameter of 0.02 for training all NB models in the experiments. We evaluate the supervised data subset selection framework (SS+f_{NB}) as Problem 6 with the objective $f_{\text{NB}(\alpha=0.02)}$. As shown in Figure 3, SS+f_{NB} and FASS+f for any choice of f perform consistently superior to random sampling (RS), which outperforms the uncertainty sampling (US) at all sizes. FASS+RS is significantly outperformed by FASS+f for any f. Drastic improvement at small sizes is achieved by SS+f_{NB}, which is outperformed by FASS+f_{NB} at larger sizes. Comparing different f in FASS+f, f_{NB} performs the best.

Nearest Neighbor Classifier: Next, we focus on how the proposed approaches perform on training NN classifiers. We evaluate the supervised data subset selection framework (SS+f_{NN}) formulated as Eqn 17 with f_{NN}. Unlike NB classifier, NN is not a probabilistic model. We model the posterior probability of a sample u given by a labeled training set S as

$$p(y|x_u; \theta(S)) = \frac{\exp\left(\max_{j\in S\cap V^y} w(u,j)\right)}{\sum_{y'\in\mathcal{Y}} \exp\left(\max_{j\in S\cap V^{y'}} w(u,j)\right)}. \tag{29}$$

As shown in Figure 4, SS+f_{NN} yields superior performance across the board over all other subset selection methods. The performances of different FASS schemes are ordered as $f_{\text{fac}} > f_{\text{NN}} > f_{\text{NB}} > f_{fs}$. Superior results are achieved with f_{fac} or f_{NN}, either of which matches well with the Nearest Neighbor classifier. Between f_{NN} and f_{fac}, f_{fac} always performs better, which may be due to the fact that the effectiveness of f_{NN} is very sensitive to the accuracy of the hypothesized class labels on which it is defined.

Logistic Regression Classifier: Lastly, we extend to select data for training an LR classifier, which is formulated and solved by the LIBLINEAR tools (Fan et al., 2008). The results are shown in Figure 5. Although f_{NN} and f_{NB} are not derived based on the LR model, superior results are still observed with them in the supervised setting. Between f_{NN} and f_{NB}, f_{NN} performs better, which indicates that f_{NN} may fit better with the properties of the LR classifiers. Similar to the results in the NN classifier, FASS with f_{NN} and f_{fac} perform better than other objectives, and yield performance competitive with SS+$\{f_{\text{NN}}, f_{\text{NB}}\}$ at large subset sizes.

[1] Data is obtained at http://qwone.com/∼jason/20Newsgroups/

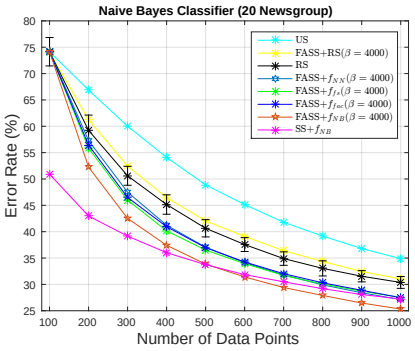

Figure 3.

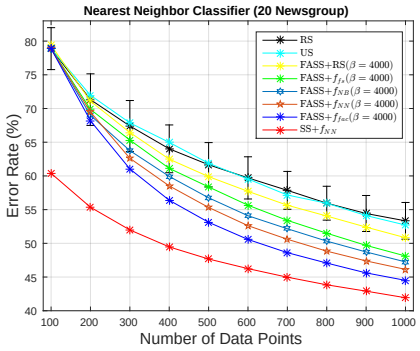

Figure 4.

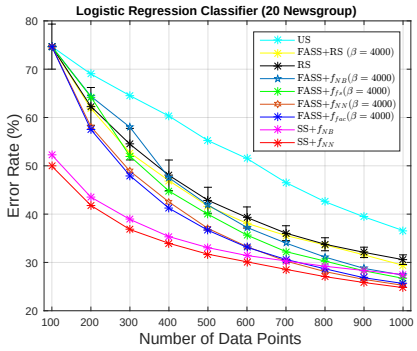

Figure 5.

Figure 6. Text categorization: classification error evaluated on (1) NB classifier; (2) NN classifier; and (3) LR classifier for different subset sizes chosen by uncertainty sampling (US), random sampling (RS) (error bars indicate standard deviation over multiple random draws), FASS with f_{fs}, f_{fac}, f_{NB}, f_{NN}, supervised data subset selection (SS) with f_{NB} or f_{NN} (SS+$\{f_{\text{NB}}, f_{\text{NN}}\}$). Error rates for NB, NN, and LR classifiers trained on the whole set are 11.1%, 19.1%, and 11.7%.

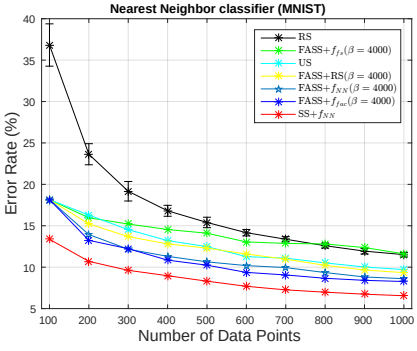

Figure 7.

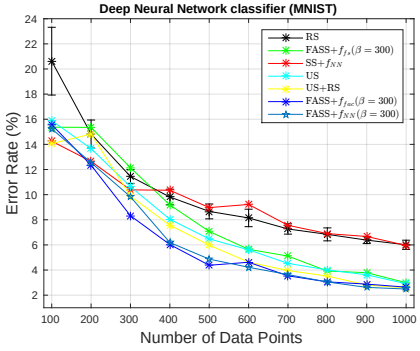

Figure 8.

Figure 9. Handwritten digit recognition: classification error evaluated on (5) NN classifier and (6) DNN classifier for various subset sizes chosen by different methods. The error rates for NN and DNN classifiers trained on the whole set are 3.1% and 1.0%.

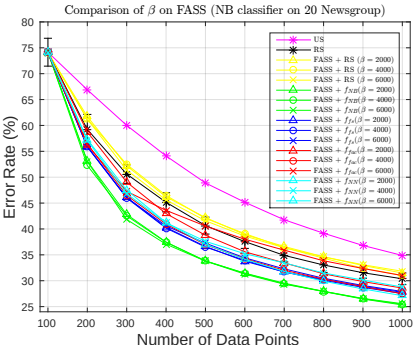

Figure 10.

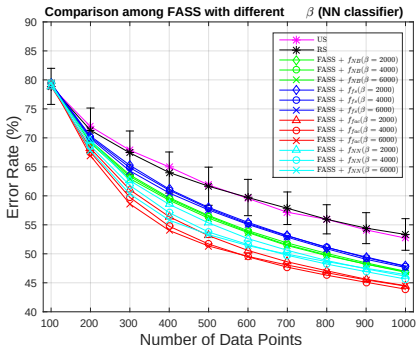

Figure 11.

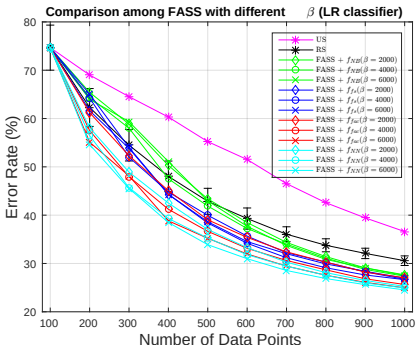

Figure 12.

Figure 13. Comparison among FASS with different β on text categorization experiments.

5.2 Handwritten Digit Recognition Experiments:

Experimental Setup: We evaluate the handwritten digit recognition task on the MNIST database [2], which consists of 60,000 training and 10,000 test samples. Each data sample is an image of a handwritten digit, which has been size-normalized and centered. The training and test data are both almost evenly divided among 10 different classes. The goal is to classify each image as a digit. Different from the setup for the text categorization experiments, we only run one instance of each subset selection strategy except for the random sampling baseline, since the training and test data are fixed. We run 10 instances of random draw for the random sampling baseline, and report the averaged

[2]The data set is downloaded from yann.lecun.com/exdb/mnist

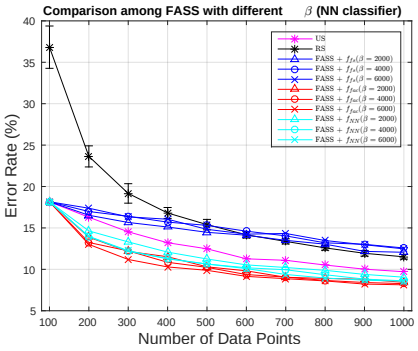

Figure 14.

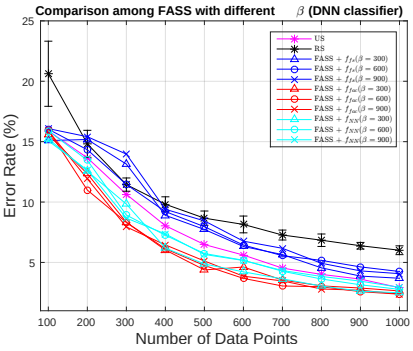

Figure 15.

Figure 16. Comparison among FASS with different β on handwritten digit recognition experiments.

classification error as its performance. For mini-batch active learning, we also experiment with $B = 100$, $T = 10$ and $k = 1000$. We bootstrap the mini-batch active learning with a different strategy: instead of randomly selecting B examples, we label a set of B representative data instances by solving $\max_{|S|=B;S\subseteq\mathcal{U}} f_{\text{fac}}(S)$ (f_{fac} does not assume any labels). On this task, we examine how various subset selection strategies perform on NN and DNN classifiers. The NB classifier is not included since it does not fit with this task. We didn't evaluate SS+f_{NB} or FASS+f_{NB} for comparison either, since the proposed Naïve Bayes submodular function f_{NB} is defined on discrete features, i.e., a set features that take categorical values.

Nearest Neighbor Classifier: First, we evaluate the proposed framework on NN classifiers. The similarity between any pair of data instances i and j is measured as $d - \|x^i - x^j\|_2^2$, where $d = \max_{u,u'\in V} \|x^u - x^{u'}\|_2^2$. We represent the feature x^i of each image i as the vector of its pixel values. We compare FASS among different submodular objectives again under the choice of $\beta = 4,000$. The results in Figure 7 show similar trends to the 20 Newsgroup experiments under the NN classifier.

Deep Neural Network Classifier: Lastly we test on DNN based classifiers. A DNN model, which consists of two convolution layers followed by two fully connected layers, is trained using Caffe (Jia et al., 2014) on the set of labeled images selected by each approach. We report the results in Figure 8. SS+f_{NN} performs well at very small sizes and then matches the random baseline. This indicates that f_{NN}, though performing well on NN and LR classifiers for the supervised setting, does not fit with the properties of the DNN model. Interestingly, drastic improvements are achieved by the simple uncertainty sampling strategy, which suggests that manually labeling the data instances that are uncertain to the current system is very valuable for updating the DNN model. Different from other classifiers, FASS+f tends to perform well when β is chosen to be small and in the range $[300, 1000]$. Here we show results for $\beta = 300$. Significant improvements are achieved by FASS+f_{NN} or f_{fac}. Though formal analysis for the DNN model is not available, the empirical results suggest that it is beneficial to select a set of uncertain data instances that are representative about the whole set as well.

5.3 Choice of β for FASS

In this part, we discuss the interplay of the choice of β for FASS schemes and its performance. We show the comparison for NB, NN, and LR classifiers on text categorization experiments in Figure 10, 11, and 12, respectively. Figure 10 and 11 show that FASS schemes are, in general, not sensitive to the choice of β when β ranges between $[2000, 6000]$ under NB and NN classifiers. In Figure 12, we observe that the performance of FASS+f_{fac} varies as different choices of β. However, consistent and significant improvements are achieved with each choice of β for FASS+f_{fac} over the random baseline. Similarly, we observe that FASS with different submodular objectives are not sensitive to β under either NN (Figure 14) or DNN (Figure 15) classifier for the handwritten digit recognition task.

6 Discussion

In this paper, we proposed a principled approach for data subset selection and active learning, for the Naïve Bayes, and the Nearest Neighbor classifiers, and linked them to submodular optimization. As an extension to our work, we would like to look at the likelihood, or more generally risk, of a large family of classifiers as a function of subsets of training data and from the perspective of submodularity. Given the enormous empirical success of deep neural networks (DNNs), it would also be useful to complement our currently empirical-only results of Figure 8 that, while showing good performance, could be significantly improved with a submodular function that better matches the properties of the DNN.

Acknowledgments: We thank Rahul Kidambi, Shengjie Wang, Chandrashekhar Lavania, and other MELODI-lab members for useful discussions. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1162606, the National Institutes of Health under award R01GM103544, and by a Google, a Microsoft, and an Intel research award. R. Iyer acknowledges support from the Microsoft Research Ph.D Fellowship.

References

Agarwal, Pankaj K, Har-Peled, Sariel, and Varadarajan, Kasturi R. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

Ahmed, S. and Atamtürk, A. Maximizing a class of submodular utility functions. *Math. Program., Ser. A*, 2009.

Bhatia, Nitin et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

Boiman, Oren, Shechtman, Eli, and Irani, Michal. In defense of nearest-neighbor based image classification. In *CVPR*, pp. 1–8. IEEE, 2008.

Chen, Yuxin and Krause, Andreas. Near-optimal batch mode active learning and adaptive submodular optimization. In *ICML*, pp. 160–168, 2013.

Cuong, Nguyen Viet, Lee, Wee Sun, and Ye, Nan. Near-optimal adaptive pool-based active learning with general loss. *UAI*, 2010.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

Feige, U. A threshold of ln n for approximating set cover. *JACM*, 1998.

Feldman, Dan, Faulkner, Matthew, and Krause, Andreas. Scalable training of mixture models via coresets. In *NIPS*, 2011.

Fisher, M.L., Nemhauser, G.L., and Wolsey, L.A. An analysis of approximations for maximizing submodular set functions—ii. *Polyhedral combinatorics*, 1978.

Gabillon, Victor, Kveton, Branislav, Wen, Zheng, Eriksson, Brian, and Muthukrishnan, S. Adaptive submodular maximization in bandit setting. In *NIPS*, 2013.

Golovin, Daniel and Krause, Andreas. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, 2010.

Gomes, Ryan and Krause, Andreas. Budgeted nonparametric learning from data streams. In *ICML*, 2010.

Guillory, Andrew and Bilmes, Jeff. Interactive submodular set cover. *ICML*, 2010.

Guillory, Andrew and Bilmes, Jeff. Active semi-supervised learning using submodular functions. In *UAI*, 2011.

Har-Peled, Sariel and Mazumdar, Soham. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300. ACM, 2004.

Hoi, Steven CH, Jin, Rong, Zhu, Jianke, and Lyu, Michael R. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

Huang, Sheng-Jun, Jin, Rong, and Zhou, Zhi-Hua. Active learning by querying informative and representative examples. In *NIPS*, 2010.

Iyer, R. and Bilmes, J. Algorithms for approximate minimization of the difference between submodular functions, with applications. *In UAI*, 2012.

Iyer, R. and Bilmes, J. Submodular optimization with submodular cover and submodular knapsack constraints. In *NIPS*, 2013.

Iyer, R., Jegelka, S., and Bilmes, J. Fast semidifferential based submodular function optimization. In *ICML*, 2013.

Iyer, Rishabh, Jegelka, Stefanie, and Bilmes, Jeff. Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization. 2014.

Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, Jonathan, Girshick, Ross, Guadarrama, Sergio, and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Kempe, David, Kleinberg, Jon, and Tardos, Éva. Maximizing the spread of influence through a social network. In *Proc. SIGKDD*, pp. 137–146. ACM, 2003.

Kirchhoff, Katrin and Bilmes, Jeff. Submodularity for data selection in machine translation. In *EMNLP*, October 2014.

Krause, A., Singh, A., and Guestrin, C. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

Lang, Ken. Newsweeder: Learning to filter netnews. In *ICML*, 1995.

Lewis, David D and Gale, William A. A sequential algorithm for training text classifiers. In *Proc. SIGIR*, pp. 3–12. Springer-Verlag New York, Inc., 1994.

Lin, Hui and Bilmes, Jeff. A class of submodular functions for document summarization. In *ALC/HLT*, 2011.

Liu, Yuzong, Wei, Kai, Kirchhoff, Katrin, Song, Yisong, and Bilmes, Jeff. Submodular feature selection for high-dimensional acoustic score spaces. In *ICASSP*, 2013.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, 1978.

Mirchandani, Pitu B and Francis, Richard L. *Discrete Location Theory*. Wiley, 1990.

Mirzasoleiman, Baharan, Karbasi, Amin, Sarkar, Rik, and Krause, Andreas. Distributed submodular maximization: Identifying representative elements in massive data. In *NIPS*, 2013.

Narasimhan, Mukund and Bilmes, Jeff. A submodular-supermodular procedure with applications to discriminative structure learning. In *UAI*, 2005.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

Nguyen, Hieu T and Smeulders, Arnold. Active learning using pre-clustering. In *ICML*, pp. 79. ACM, 2004.

Nigam, Kamal, McCallum, Andrew Kachites, Thrun, Sebastian, and Mitchell, Tom. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

Prasad, Adarsh, Jegelka, Stefanie, and Batra, Dhruv. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. In *NIPS*, 2014.

Reed, Colorado and Ghahramani, Zoubin. Scaling the indian buffet process via submodular maximization. *arXiv preprint arXiv:1304.3285*, 2013.

Settles, Burr. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

Seung, H Sebastian, Opper, Manfred, and Sompolinsky, Haim. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294. ACM, 1992.

Shah, Ronak, Iyer, Rishabh, and Chaudhuri, Subhasis. Object mining for large video data. *Proc. BMVC*, 22(10): 761–767, 2011.

Shamaiah, Manohar, Banerjee, Siddhartha, and Vikalo, Haris. Greedy sensor selection: Leveraging submodularity. In *Proc. CDC*, pp. 2572–2577. IEEE, 2010.

Shinohara, Yusuke. A submodular optimization approach to sentence set selection. In *ICASSP*, pp. 4112–4115. IEEE, 2014.

Shioura, Akiyoshi. On the pipage rounding algorithm for submodular function maximizationa view from discrete convex analysis. *Discrete Mathematics, Algorithms and Applications*, 1(01):1–23, 2009.

Singla, Adish, Bogunovic, Ilija, Bartók, Gábor, Karbasi, Amin, and Krause, Andreas. Near-optimally teaching the crowd to classify. *arXiv preprint arXiv:1402.2092*, 2014.

Stobbe, P. and Krause, A. Efficient minimization of decomposable submodular functions. In *NIPS*, 2010.

Tsang, Ivor W, Kwok, James T, and Cheung, Pak-Ming. Core vector machines: Fast svm training on very large data sets. In *JMLR*, 2005.

Tschiatschek, Sebastian, Iyer, Rishabh K, Wei, Haochen, and Bilmes, Jeff A. Learning mixtures of submodular functions for image collection summarization. In *NIPS*, 2014.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning: Extended version. 2015.

Wei, Kai, Liu, Yuzong, Kirchhoff, Katrin, and Bilmes, Jeff. Using document summarization techniques for speech data subset selection. In *NAACL/HLT*, 2013.

Wei, Kai, Iyer, Rishabh, and Bilmes, Jeff. Fast multi-stage submodular maximization. In *ICML*, 2014a.

Wei, Kai, Liu, Yuzong, Kirchhoff, Katrin, Bartels, Chris, and Bilmes, Jeff. Submodular subset selection for large-scale speech training data. In *ICASSP*, 2014b.

Wei, Kai, Liu, Yuzong, Kirchhoff, Katrin, and Bilmes, Jeff. Unsupervised submodular subset selection for speech data. In *ICASSP*, 2014c.

Xu, Zhao, Yu, Kai, Tresp, Volker, Xu, Xiaowei, and Wang, Jizhi. *Representative sampling for text classification using support vector machines*. Springer, 2003.

Zheng, Jingjing, Jiang, Zhuolin, Chellappa, Rama, and Phillips, Jonathon P. Submodular attribute selection for action recognition in video. In *NIPS*, 2014.