Distributed Maximization of “Submodular plus Diversity” Functions for Multi-label Feature Selection on Huge Datasets

Mehdrad Ghadiri **Mark Schmidt**
University of British Columbia

Abstract

There are many problems in machine learning and data mining which are equivalent to selecting a non-redundant, high “quality” set of objects. Recommender systems, feature selection, and data summarization are among many applications of this. In this paper, we consider this problem as an optimization problem that seeks to maximize the sum of a sum-sum diversity function and a non-negative monotone submodular function. The diversity function addresses the redundancy, and the submodular function controls the predictive quality. We consider the problem in big data settings (in other words, distributed and streaming settings) where the data cannot be stored on a single machine or the process time is too high for a single machine. We show that a greedy algorithm achieves a constant factor approximation of the optimal solution in these settings. Moreover, we formulate the multi-label feature selection problem as such an optimization problem. This formulation combined with our algorithm leads to the first distributed multi-label feature selection method. We compare the performance of this method with centralized multi-label feature selection methods in the literature, and we show that its performance is comparable or in some cases is even better than current centralized multi-label feature selection methods.

1 Introduction

Many problems from different areas of machine learning and data mining can be modeled as an optimization problem that tries to maximize the sum of a sum-sum diversity function (which is the sum of the distances between all of the pairs in a given subset) and a non-negative monotone submodular function. Examples include query diversification problem in the area of databases [Demidova et al., 2010, Liu et al., 2009], search result diversification [Agrawal et al., 2009, Drosou and Pitoura, 2010], and recommender systems [Yu et al., 2009]. The size of the datasets in these applications is growing rapidly, and there is a need for scalable methods to tackle these problems on huge datasets. Inspired by these applications, we propose an algorithm for approximately solving this optimization problem with a theoretical guarantee in distributed and steaming settings. Borodin et al. [2017] presented a 0.5-approximation for this optimization problem in the centralized setting in which data can be stored and processed on a single machine. In this paper, we consider this problem for big data settings where the data cannot be stored on a single machine, or the process time is too high for a single machine. We show that our algorithm achieves a 1/31-approximation. Note that solving this problem in a distributed or streaming setting is strictly harder than solving it in the centralized setting because, in the aforementioned settings, the algorithm does not use all of the data. As a result, our algorithm is $\frac{\sqrt{d/k}}{2}$ times faster in the distributed setting and it needs $\sqrt{d/k}$ times less memory in the streaming setting compared to the centralized setting, where d is the size of the ground set (for example, the number of features in the feature selection problem), and k is the number of machines (in the distributed setting) or is the number of partitions of the data (in the streaming setting). Therefore, our algorithm gives a worse approximate solution compared to the centralized method of Borodin et al. [2017] but it is much faster and needs less memory. This trade-off might be interesting and useful in some applications.

One of the problems that can be modeled as such an optimization problem and is in need of scalable methods in modern applications is multi-label feature selection. The diversity part controls the redundancy of the se-

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

lected features and the submodular part is to promote features that are relevant to the labels. A multi-label dataset is made up of a number of samples, features, and labels. Each sample is a set of values for the features and labels. Usually, labels have binary values. For example, if a patient has diabetes or not. Multi-label datasets can be found in different areas, including but not limited to semantic image annotation, protein and gene function studies, and text categorization [Kashef et al., 2018]. Applications, number, and size of such datasets are growing very rapidly, and it is necessary to develop efficient and scalable methods to deal with them.

Feature selection is a fundamental problem in machine learning. Its goal is to decrease the dimensionality of a dataset in order to improve the learning accuracy, decrease the learning and prediction time, and prevent overfitting. There are three different categories of feature selection methods depending on their interaction with the learning methods. Filter methods select the features based on the intrinsic properties of the data and are totally independent of the learning method. Wrapper methods select the features according to the accuracy of a specific learning method, like SVMs. Finally, embedded methods select the features as a part of their learning procedure [Guyon and Elisseeff, 2003]. Decision trees and use of ℓ_0 and ℓ_1 regularization for feature selection fall into the latter. When the number of features is large, filter methods are a reasonable choice since they are fast, resistant to over-fitting, and independent of the learning model. Therefore, we can quickly select a number of features with filter methods and then try different learning methods to see which one fits the data better (possibly with wrapper or embedded feature selection methods). However, with millions of features, centralized filter methods are not applicable anymore. To deal with such huge datasets, we need scalable methods. Although there were efforts to develop scalable and distributed filter methods for single-label datasets [Zadeh et al., 2017, Bolón-Canedo et al., 2015a], to the best of our knowledge, there are no previous distributed multi-label feature selection method.

In this paper, we propose an information theoretic filter feature selection method for multi-label datasets that is usable in distributed, streaming, and centralized settings. In the centralized setting, all of the data is stored and can be processed on a single machine. In the distributed setting, the data is stored on multiple machines, and there is no shared memory between machines. In the streaming setting, although the computation is done on a single machine, this machine does not have enough memory to store all of the data at once. The data in our method is distributed vertically which means that the features are distributed between machines instead of samples (horizontal distribution). Feature selection is considered harder when the data is distributed vertically because we lose much information about the relations of the features [Bolón-Canedo et al., 2015b]. However, when the number of instances is small, and the number of features is large (for example, biological or medical datasets) vertical distribution is the only reasonable choice. Our work can be seen as an extension of Borodin et al. [2017] to distributed and streaming settings or an extension of Zadeh et al. [2017] to multi-label data. However, our results cannot be derived from these previous works in a straightforward manner. The main contributions of the paper are listed in the following.

Our Contributions

- We present a greedy algorithm for maximizing the sum of a sum-sum diversity function and a non-negative monotone submodular function in the distributed and streaming settings. We prove that it achieves a constant factor approximation of the optimal solution.

- We formulate the multi-label feature selection problem as such a combinatorial optimization problem. Using this formulation we present information theoretic filter feature selection methods for distributed, steaming, and centralized settings. The distributed method is the first distributed multi-label feature selection method proposed in the literature.

- We perform an empirical study of the proposed distributed method and compare its results to different centralized multi-label feature selection methods. We show that the results of the distributed method are comparable to the current centralized methods in the literature. We also compare the runtime and the value of the objective function that our centralized and distributed methods achieve. Note that the centralized methods have access to the all of the data and can do computation on it. We do not expect that our distributed or streaming method to beat the centralized methods because it is not possible. However, we argue that our results are comparable to the results of centralized methods and our method is much faster (in case of the distributed setting) and needs much less memory (in case of the streaming setting). We compared our results with the centralized methods (this comparison is unfair to the distributed setting) in the literature because to the best of our knowledge there is no distributed multi-label feature selection method prior to this work.

Our techniques can be used prior to multi-label classification, multi-label regression, and in some multi-task learning setups. The structure of the paper is as follows. In the next section, we review the related work and preliminaries. In Section 3, we formulate the multi-label feature selection problem as the mentioned optimization problem and present the algorithm for maximizing it in the distributed and streaming settings. In Section 4, we show the theoretical approximation guarantee of the proposed algorithm. In Section 5, we evaluate the performance of the proposed distributed algorithm in practice.

2 Related Work

In this section, we review the previous works on different aspects of the problem including diversity maximization, submodular maximization, composable core-sets, and feature selection.

Diversity Maximization and Submodular Maximization

Usually, the diversity maximization problem is defined on a metric space of a set of points U with the goal of finding a subset of them which maximizes a diversity function subject to a constraint. For example, a cardinality constraint or a matroid constraint. If S is a subset of the points, the sum-sum diversity of S is $D(S) = 0.5\sum_{x\in S}\sum_{y\in S} d(x,y)$ where $d(.,.)$ is a metric distance. In the centralized setting, a simple greedy or local search algorithm can achieve a half approximation of the optimal solution subject to $|S| = k$ [Hassin et al., 1997, Abbassi et al., 2013]. TA better approximation factor is not achievable under the planted clique conjecture [Bhaskara et al., 2016, Borodin et al., 2017].

Submodular functions are important concepts in machine learning and data mining with many applications. See Krause and Guestrin [2008] for their applications. A submodular function is a set function with a diminishing marginal gain. A function $f : 2^U \to \mathbb{R}$ is submodular if $f(A\cup\{x\}) - f(A) \geq f(B\cup\{x\}) - f(B)$ for any $A \subseteq B \subset U$, and $x \in U \setminus B$. It is monotone if $f(A) \leq f(B)$ and it is non-negative if $f(A) \geq 0$ for any $A \subseteq B \subseteq U$. Maximizing a monotone submodular function subject to a cardinality constraint is NP-hard but using a simple greedy algorithm we can achieve $(1-\frac{1}{e})$ of the optimal solution. A better approximation factor is not achievable using a polynomial time algorithm unless P=NP [Krause and Golovin, 2014].

Let U be a set and $f(.)$ be a submodular function defined on U and $d(.,.)$ be a metric distance defined between pairs of elements of U. Borodin et al. [2017] showed that in the centralized setting, using a simple greedy algorithm, we can achieve half of the optimal value for maximizing $f(S) + \lambda\sum_{\{u,v\}:u,v\in S} d(u,v)$ subject to $S \subseteq U$ and $|S| = k$. This result is extended to semi-metric distances in Abbasi Zadeh and Ghadiri [2015]. Similar problems are considered in Dasgupta et al. [2013] where the diversity part can be other diversity functions. Namely, they considered the sum-sum diversity, the minimum spanning tree, and the minimum of distances between all pairs. They showed that the greedy algorithm achieves a constant factor approximation in all of these cases.

Composable Core-sets

In computational geometry, a core-set is a small subset of points that approximately preserve a measure of the original set [Agarwal et al., 2005]. Composable core-sets extend this property to the combination of sets. Therefore, they can be used in a divide and conquer manner to find an approximate solution. Let U be a set, $f : 2^U \to \mathbb{R}$ be a set function on U, $(T^1,\dots,T^m)$ be a random partitioning of elements of U, and k be a positive integer. Let $\texttt{OPT}(T) = \arg\max_{S\subseteq T,|S|=k} f(S)$ where $T \subseteq U$. Let $\texttt{ALG}$ be an algorithm which takes $T \subseteq U$ as an input and outputs $S \subseteq T$. For $\alpha > 0$, we call $\texttt{ALG}$ an α-approximate composable core-set with size k for f if the size of its output is k and $f(\texttt{OPT}(\texttt{ALG}(T^1)\cup\cdots\cup\texttt{ALG}(T^m))) \geq \alpha f(\texttt{OPT}(T^1\cup\cdots\cup T^m))$ [Indyk et al., 2014]. We call $\texttt{ALG}$ an α-approximate *randomized* composable core-set with size k for f if the size of its output is k and $\mathbb{E}[f(\texttt{OPT}(\texttt{ALG}(T^1)\cup\cdots\cup\texttt{ALG}(T^m)))] \geq \alpha f(\texttt{OPT}(T^1\cup\cdots\cup T^m))$ [Mirrokni and Zadimoghaddam, 2015]. Composable core-sets and randomized composable core-sets can be used in distributed settings (like the MapReduce framework) and streaming settings (see Figure 1).

Composable core-sets first were used to approximately solve several diversity maximization problems in distributed and streaming settings [Indyk et al., 2014]. It resulted in an approximation algorithm for the sum-sum diversity maximization with an approximation factor of less than 0.01. This approximation factor is improved to $\frac{1}{12}$ in Aghamolaei et al. [2015]. Randomized composable core-sets were first introduced to tackle submodular maximization problem in distributed and streaming settings which resulted in a 0.27-approximation algorithm for monotone submodular functions [Mirrokni and Zadimoghaddam, 2015]. Then they were used to improve the approximation factor of the sum-sum diversity maximization from $\frac{1}{12}$ to 0.25 [Zadeh et al., 2017]. The randomized composable core-sets used in the latter case find the approximate solution with high probability instead of expectation.

There are a number of other works on distributed submodular maximization [Mirzasoleiman et al., 2016, Barbosa et al., 2015]. Moreover, submodular and weak submodular functions are used for distributed *single-label* feature selection [Khanna et al., 2017]. We should note that the discussed objective function in our work is neither submodular nor weak submodular. This is because of the diversity term of the function. An advantage of using this diversity function is that it is evaluated by a pairwise distance function. As a result, it is easy to evaluate our objective function on datasets with few samples. On the contrary, evaluating the pure submodular functions, that were used for feature selection in the literature, are quite hard and need a large amount of data and computing power.

Feature Selection and Multi-label Feature Selection

Filter feature selection methods select features independent of the learning algorithm. Hence, they are usually faster and immune to overfitting [Guyon and Elisseeff, 2003]. Mutual information based methods are a well-known family of filter methods. The best-known method of this kind for single-label feature selection is minimum redundancy and maximum relevance (mRMR) which tries to find a subset of features S that maximizes the following objective function using a greedy algorithm

$$\frac{1}{|S|}\sum_{x_i\in S} I(x_i,c) - \frac{1}{|S|^2}\sum_{x_i,x_j\in S} I(x_i,x_j),$$

where $I(.,.)$ is the mutual information function, and c is the label vector [Peng et al., 2005]. The proposed method in this paper can be seen as a variation of mRMR which is capable of being used for multi-label feature selection in distributed, streaming, and centralized settings.

Although there have been great advancements in centralized feature selection, there are few works on distributed feature selection, and most of them distribute the data horizontally. Zadeh et al. [2017] was the first work on the single-label vertically distributed feature selection that considered the redundancy of the features. Their method selects features using randomized composable core-sets in order to maximize a diversity function defined on the features. Although there are some similarities between the formulations presented in Zadeh et al. [2017] and this work, we should note that the single-label formulation cannot be applied directly to multi-label datasets. Moreover, maximization of the functions and the analysis of the algorithms to prove the theoretical guarantee are completely different.

Most of the multi-label feature selection methods transform the data to a single-label form. Binary relevance (BR) and label powerset (LP) are two common ways to do so. BR methods consider each label separately and use a single-label feature selection method to select features for each label, and then they aggregate the selected features. A disadvantage of BR methods is that they cannot consider the relations of the labels. LP methods consider the multi-label dataset as one single-label multi-class dataset where each class of its single label are a possible combination of labels in the dataset (treating the labels as a binary string). Then they apply a single-label feature selection method. Although LP methods consider the relations of the labels, they have significant drawbacks. For example, some classes may end up with very few samples or none at all. Moreover, the method is biased toward the combination of the labels which exist in the training set [Kashef et al., 2018]. Our proposed method does not transform the data to single-label data and is designed in a way to not suffer from the mentioned disadvantages.

3 Problem Formulation

Let U be a set of d features and L be a set of t labels. We also have a set A of n instances each of which is a vector of observations for elements of $U \cup L$. The goal of *multi-label feature selection* is to find a small *non-redundant* subset of U which can *predict* labels in L accurately. In order to quantify redundancy it is natural to use a metric distance d over the feature set to measure dissimilarity. In our application (feature selection) we particularly are interested in the following metric distance. For any $u_i, u_j \in U$, we define

$$\begin{aligned} d(u_i,u_j) &= 1 - \frac{I(u_i,u_j)}{H(u_i,u_j)} \\ &= 1 - \frac{\sum_{x\in u_i, y\in u_j} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}}{-\sum_{x\in u_i,y\in u_j} p(x,y)\log p(x,y)}, \end{aligned}$$

where $H(.,.)$ is the joint entropy and $I(.,.)$ is the mutual information. This distance function is called *normalized* (values lie between 0 and 1) *variation of information* and it is a metric [Nguyen et al., 2010]. In Zadeh et al. [2017], this distance function plus a modular function is used for single-label feature selection.

In order to quantify the predictive quality of the selected features, we define a non-negative monotone submodular function $g : 2^U \to \mathbb{R}$ which measures the relevance of the selected features to the labels. For any positive integer p, we define

$$g(S) = \sum_{\ell\in L} \underset{x\in S}{\mathrm{top}^{\mathrm{p}}}\{MI(x,\ell)\},$$

where $\mathrm{top}^{\mathrm{p}}{}_{x\in S}\{MI(x,\ell)\}$ is the sum of the p largest numbers in $\{MI(x,\ell)|x \in S\}$. Here $MI(x,\ell) = \frac{I(x,\ell)}{\sqrt{H(x)H(\ell)}}$ is the normalized mutual information where $H(.)$ is the entropy function and the value $MI(.,.)$

lies in $[0, 1]$. Note that if we only have one label (i.e., $|L| = 1$), and $p = d$ (the number of all features of the dataset) then g will be exactly the modular function used in Zadeh et al. [2017]. Therefore, our formulation is a generalization of theirs. Using the top^{p} function, this formulation tries to select at least p relevant features for each label. In order to understand the importance of top^{p} function, we discuss two extreme cases: $p = 1$ and $p = d$. If $p = 1$ then a feature that is somewhat relevant to all the features can dominate the $g(S)$ and prevent other features, that are highly relevant to one or few features, to get selected. If $p = d$ then a label that has a lot of relevant features can dominate $g(S)$ and prevent other labels to get relevant features, while a few features would be enough for predicting this label with a high accuracy. In the following lemma, we show that g has the nice properties we need in our model. Its proof is included in Appendix A.

Lemma 1. *g is a non-negative, monotone, submodular function.*

Hence if we define $f(S) = g(S) + \sum_{\{u,v\}\in S} d(u, v)$, then our feature selection model reduces to solving the following combinatorial optimization problem.

$$\max_{\substack{S\subseteq U\\ |S|=k}} f(S) = \max_{\substack{S\subseteq U\\ |S|=k}} \{g(S) + \sum_{\{u,v\}\in S} d(u, v)\}, \qquad (1)$$

where $d(.,.)$ is a metric distance and $g(.)$ is a non-negative monotone submodular function. In the actual feature selection method we are free to scale the relative contributions of the diversity or submodular parts, since both metric and submodular functions are closed under multiplication by a positive constant. Hence, we use a weighted version of the objective function in our application.

Algorithm 1: Greedy

```
Input: Set of features $U$, set of labels $L$, number of
  features we want to select $k$.
Output: Set $S \subset U$ with $|S| = k$.
$S \leftarrow \{\arg\max_{u\in U} g(\{u\})\}$;
forall $2 \le i \le k$ do
    $u^* \leftarrow \arg\max_{u\in U\setminus S} \; g(S \cup \{u\}) - g(S) + \sum_{x\in S} d(x, u)$;
        ▷ This arg max has a consistent tiebreaking
          rule (see Definition 1).
    Add $u^*$ to $S$;
Return $S$;
```

The problem (1) is NP-hard but Borodin et al. [2017] show that Algorithm 2 is a half approximation in the centralized setting. Note that this is a greedy algorithm under the objective where $g(S)$ is scaled by $\frac{1}{2}$. On the other hand, Algorithm 1 is a standard greedy algorithm for (1) and in the next section we show it is a constant factor randomized composable core-set for any functions f which are the sum of a sum-sum diversity function and a non-negative, monotone, submodular function. Combining these we conclude that Algorithm 3 is a constant factor approximation algorithm for maximizing f. Moreover, Algorithm 3 can be used both in distributed and streaming settings, as illustrated in Figure 1. In our experiments, to select k features, we use the following function.

$$h(S) = (1-\lambda)\frac{k(k-1)}{2p|L|} g(S) + \lambda \sum_{x_i,x_j\in S} d(x_i, x_j). \quad (2)$$

As discussed, the first term of $h(S)$ controls redundancy of the selected features and the second term is to promote features that are relevant to the labels. The term $\frac{k(k-1)}{2p|L|}$ is a normalization coefficient to make the range of both terms the same. Also, λ is a hyper-parameter which controls the effect of two criteria on the final function.

Algorithm 2: AltGreedy

```
Input: Set of features $U$, set of labels $L$, number of
  features we want to select $k$.
Output: Set $S \subset U$ with $|S| = k$.
$S \leftarrow \{\arg\max_{u\in U} g(\{u\})\}$;
forall $2 \le i \le k$ do
    $u^* \leftarrow \arg\max_{u\in U\setminus S} \; \frac{1}{2}(g(S \cup \{u\}) - g(S)) + \sum_{x\in S} d(x, u)$;
    Add $u^*$ to $S$;
Return $S$;
```

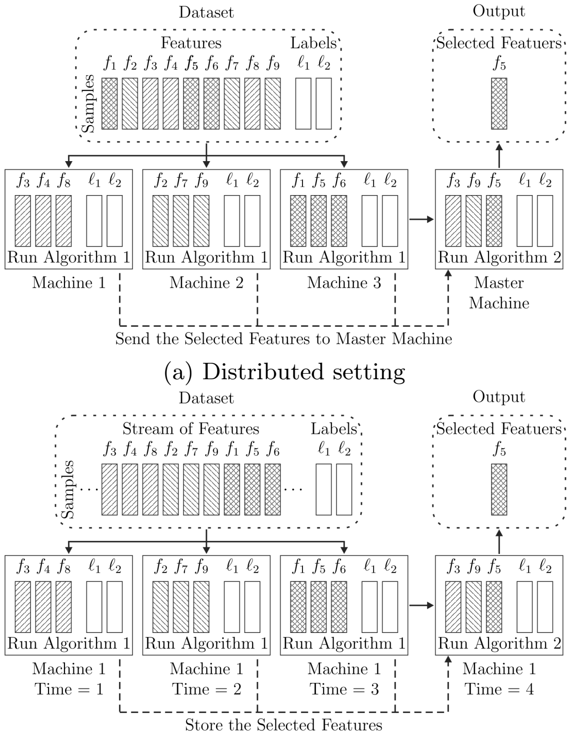
4 Theoretical Results

Let $f(S) = D(S)+g(S)$ be a set function defined on 2^U where $g(S)$ is a non-negative, monotone, submodular function and $D(S)$ is a sum-sum diversity function, i.e. $D(S) = \sum_{\{u,v\}\in S} d(u, v)$ where $d(.,.)$ is a metric distance. In this section, we show that Algorithm 1 is a constant factor randomized composable core-set with size k for f. We also show that running Algorithm 3 which is equivalent to running Algorithm 1 in each slave machine and then running Algorithm 2 in the master machine on the union of outputs of slave machines is a constant factor randomized approximation algorithm for maximizing f subject to a cardinality constraint.

Algorithm 3: Distributed Greedy

```
Input: Set of features $U$, set of labels $L$, number of
  features we want to select $k$, number of machines $m$.
Output: Set $S \subset U$ with $|S| = k$.
Randomly partition $U$ into $(T_i)_{i=1}^m$;
forall $1 \le i \le m$ do
    $S_i \leftarrow$ output of $Greedy(T_i, L, k)$;
$S \leftarrow$ output of $AltGreedy(\cup_{i=1}^m S_i, L, k)$;
Return $S$;
```

We use the following key concept of a β-nice algorithm from Mirrokni and Zadimoghaddam [2015] throughout our analysis.


Dataset
Features
Labels
Samples
f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9
ℓ_1 ℓ_2
Output
Selected Featuers
f_5
f_3 f_4 f_8 ℓ_1 ℓ_2
f_2 f_7 f_9 ℓ_1 ℓ_2
f_1 f_5 f_6 ℓ_1 ℓ_2
f_3 f_9 f_5 ℓ_1 ℓ_2
Run Algorithm 1
Run Algorithm 1
Run Algorithm 1
Run Algorithm 2
Machine 1
Machine 2
Machine 3
Master Machine
Send the Selected Features to Master Machine

(a) Distributed setting

Dataset
Stream of Features
Labels
Samples
f_3 f_4 f_8 f_2 f_7 f_9 f_1 f_5 f_6
ℓ_1 ℓ_2
Output
Selected Featuers
f_5
f_3 f_4 f_8 ℓ_1 ℓ_2
f_2 f_7 f_9 ℓ_1 ℓ_2
f_1 f_5 f_6 ℓ_1 ℓ_2
f_3 f_9 f_5 ℓ_1 ℓ_2
Run Algorithm 1
Run Algorithm 1
Run Algorithm 1
Run Algorithm 2
Machine 1 Time = 1
Machine 1 Time = 2
Machine 1 Time = 3
Machine 1 Time = 4
Store the Selected Features

(b) Streaming setting

Figure 1: Algorithm 3 operating in big data settings.

Definition 1. Let f be a set function on 2^U. Let $\texttt{ALG}$ be an algorithm that given any $T \subseteq U$ outputs $\texttt{ALG}(T) \subseteq T$. Let $t \in T \setminus \texttt{ALG}(T)$. For $\beta \in \mathbb{R}^+$, we call $\texttt{ALG}$ a β-nice algorithm if it has the following properties.

- $\texttt{ALG}(T) = \texttt{ALG}(T \setminus \{t\})$.
- $f(\texttt{ALG}(T) \cup \{t\}) - f(\texttt{ALG}(T)) \leq \beta \frac{f(\texttt{ALG}(T))}{k}$.

The intuition behind the first condition is simply that by removing an element of T which is not used in the algorithm's output, we do not change the output. This is effectively a condition on how we perform tiebreaking. The second condition helps to bound $f(\texttt{ALG}(T) \cup O)$ where O is a global optima. Our theoretical analysis heavily relies on the following theorem which is proved in Appendix B.

Theorem 1. Let $k \geq 10$. Algorithm 1 is a 5-nice algorithm for $f(.) = D(.) + g(.)$. Also, if $\texttt{ALG}$ is Algorithm 1, $T \subseteq U$, and $t \in T \setminus \texttt{ALG}(T)$, then $\frac{4.5}{k-1} f(\texttt{ALG}(T)) \geq \sum_{x \in \texttt{ALG}(T)} d(t,x)$.

Our main result is that Algorithm 3 is a constant factor approximation algorithm.

Theorem 2. Let $k \geq 10$. Algorithm 3 gives a $\frac{1}{31}$-approximation solution in expectation for maximizing $f(S)$ subject to $|S| = k$.

Table 1: Specifications of the datasets.

Dataset Name	# Features	# Instances	# Labels
Corel5k	499	5000	374
Eurlex-ev	5000	19,348	3993
Synthesized	800	256	8

We note that for $k < 10$, the constant degrades so we focus on the large k regime. The proof of this theorem follows from two key lemmas which bound the diversity and submodular portions of an optimal solution. We use O to denote a global optimum. To state the lemmas, we need the following notations. Let $\texttt{OPT}(T) = \arg\max_{R \subseteq T} f(R)$ subject to $|R| = k$. Let U be the set of all elements (for example, the set of all features for the feature selection problem) and $(T^1, \ldots, T^m)$ be a random partitioning of the elements of U.

Lemma 2. Let $\texttt{ALG}$ be Algorithm 1 and $S^i = \texttt{ALG}(T^i)$. Then $D(O) \leq 8.5 f(\texttt{OPT}(\cup_{i=1}^m S^i))$.

Lemma 3. Let $\texttt{ALG}$ be Algorithm 1 and $S^i = \texttt{ALG}(T^i)$. Then $g(O) \leq 6 f(\texttt{OPT}(\cup_{i=1}^m S^i)) + \mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^m S^i))]$.

We use Theorem 1 and techniques from a number of papers [Zadeh et al., 2017, Indyk et al., 2014, Mirrokni and Zadimoghaddam, 2015, Aghamolaei et al., 2015] to prove these two key lemmas in Appendix B. Even in cases where some parts of proofs are similar to previous work we include a complete proof for the sake of completeness. We should note that our analysis is not a straightforward combination of the ideas in the mentioned papers. Using Lemma 2 and 3, we can easily prove Theorem 2.

Proof of Theorem 2. Lemma 2 and 3 immediately yield $f(O) \leq 15.5 \mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^m S^i))]$. Based on Borodin et al. [2017], we know that Algorithm 2 is a half approximation algorithm for maximizing f. Therefore, if $\texttt{ALG'}$ is Algorithm 2 then $f(\texttt{OPT}(\cup_{i=1}^m S^i)) \leq 2 f(\texttt{ALG'}(\cup_{i=1}^m S^i))$. Hence $f(O) \leq 31 \mathbb{E}[f(\texttt{ALG'}(\cup_{i=1}^m S^i))]$ which is exactly the statement of the theorem. □

5 Empirical Results

In this section, we investigate the performance of our method in practice. In the first experiment, we compare our distributed method with centralized multi-label feature selection methods in the literature on a classification task. We show that our method's performance is comparable to, or in some cases is even better than previous centralized methods. Next, we compare our distributed and centralized methods on two large datasets. We show that the distributed algorithm achieves almost the same objective function value and it is much faster. This implies that the distributed algorithm achieves a better approximation in practice compared to the theoretical guarantee.

Table 2: Comparison of the distributed and the centralized algorithms. "h" and "m" means hour and minute.

Dataset Name	Reference	# Features	# Instances	# Labels	# Selected Features	# Machines	Distributed Algorithm Objective Value	Centralized Algorithm Objective Value	Distributed Algorithm Runtime	Centralized Algorithm Runtime	Speed-up
RCV1V2	[Lewis et al., 2004]	47,236	6000	101	10	69	22.7	22.6	2.8m	1h 33m	33.2
					50	31	618.7	616.4	10.8m	2h 30.0m	15.1
					100	22	2468.2	2490.7	20.3m	3h 39m	10.8
					200	16	9338.7	10,016.0	47.0m	6h 16.8m	8.0
TMC2007	[Srivastava and Zane-Ulman, 2005]	49,060	28,596	22	10	71	22.8	22.6	4.6m	2h 32.5m	33.4
					50	32	620.0	615.6	24.2m	6h 24.7m	15.9
					100	23	2510.0	2487.7	59.5m	11h 6.2m	11.2
					200	16	10,104.3	10,001.4	2h 41.3m	20h 49.8m	7.7

Comparison to Centralized Methods

As mentioned in Section 2, most of the multi-label feature selection methods convert the multi-label dataset to one or multiple single-label datasets and then use single-label feature selection methods and then aggregate the results. Binary relevance (BR) and label powerset (LP) are the two best known of these conversions. Here, we combine these two conversion methods with two single-label feature selection methods which results in four different centralized feature selection methods. We considered ReliefF (RF) [Kononenko, 1994, Robnik-Sikonja and Kononenko, 2003] and information gain (IG) [Zhao et al., 2010] for single-label methods. These methods compute a score for each feature and for aggregating their results in Binary Relevance conversion, it is enough to calculate the sum of the scores of each feature and use these scores for selecting features. These methods are used before in the literature for multi-label feature selection [Chen et al., 2007, Dendamrongvit et al., 2011, Spolaor et al., 2011, Spolaôr et al., 2012, 2013].

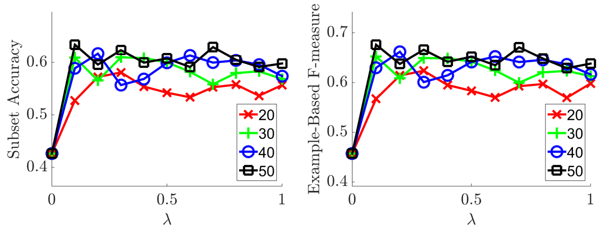


Figure 2: Effect of λ on the performance of the method.

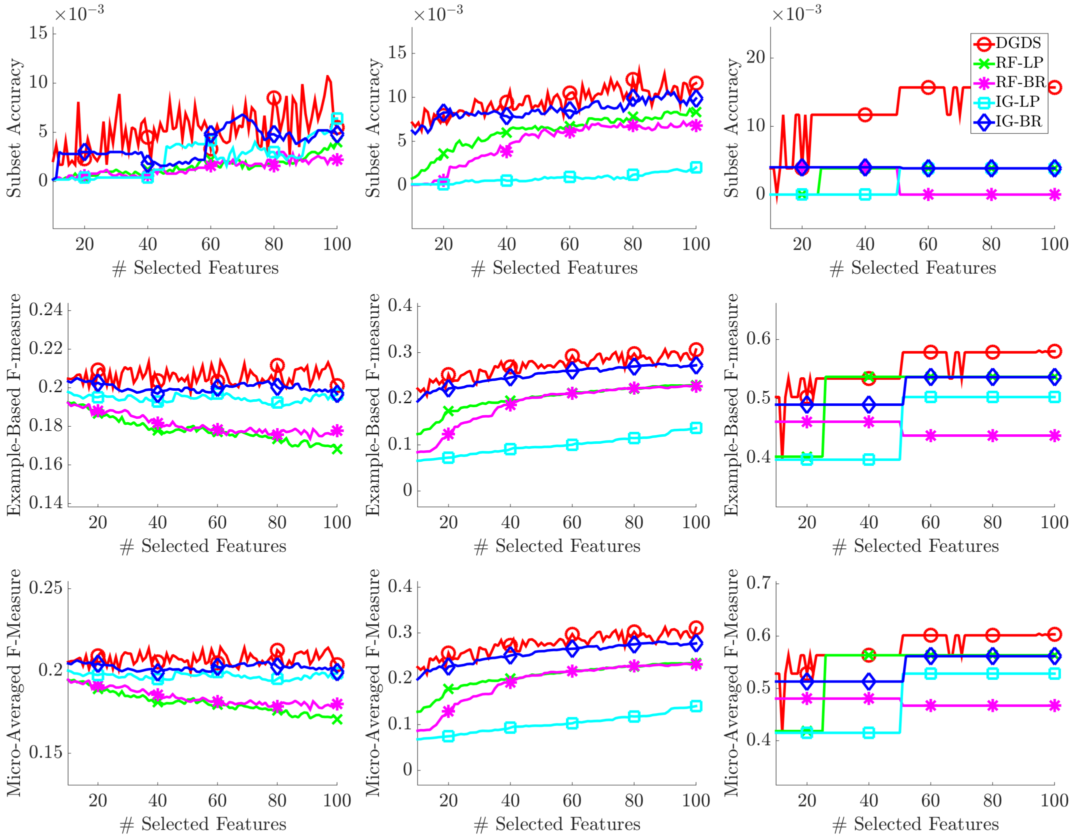
For comparison, we selected 10 to 100 features with each method and did a multi-label classification using BRKNN-b proposed in Xioufis et al. [2008]. We did a 10-fold cross validation with five neighbors for BRKNN-b. We evaluated the classification outputs over five multi-label evaluation measures. They are subset accuracy, example-based accuracy, example-based F-measure, micro-averaged F-measure, and macro-averaged F-measure [Spolaôr et al., 2013, Kashef et al., 2018]. Evaluation measures are defined in Appendix C.

We used the Mulan library for the classification and computation of the evaluation measures [Tsoumakas et al., 2011]. We used a synthesized dataset and two real-world datasets-Corel5k [Duygulu et al., 2002] and Eurlex-ev [Francesconi et al., 2010]. Their specifications are shown in table 1. The synthesized dataset made up of eight labels. Each label has two original features that repeated 50 times. One of the features has the same value as its label in half of the samples, and the other one has the same value as its label in a quarter of the samples. The results of this dataset show that our method outperforms other methods on a dataset with redundant features. The results of this experiments are shown in Figure 3. Results of example-based accuracy and macro-average F-measure comparison for these datasets are included in Appendix D. We named our method distributed greedy diversity plus submodular (DGDS) in the plots. The other methods are named based on the conversion method they use (i.e., BR or LP) and the feature selection method they use (i.e., RF or IG). In the experiments, we used $\lambda = 0.5$ and top^{10} for our method. Moreover, methods are compared on three other datasets in Appendix D. Results of the distributed method fluctuate more compared to other methods. The reason is that, for every number of features, we did the feature selection, including the random partitioning, from scratch. This caused more variation in its results but also showed that the method is relatively stable and does not produce poor quality results for different random partitionings.

As discussed, we compared our method to centralized feature selection methods because there is no distributed multi-label feature selection method prior to our work. We should note that this comparison is unfair to the distributed method because it uses much less of the data compared to centralized methods. For example, it does not use the relation (or the distance) between the features in different machines. The advantage of the distributed method is that it is much faster and scalable. This is supported by experiments on its speed-up (see Table 2).

Comparison of Distributed and Centralized Algorithms

Here, we compare the performance of our proposed algorithm (Algorithm 3) with the centralized algorithm introduced in Borodin et al. [2017] (Algorithm 2) on the optimization task. We compare the runtime and the value of the objective function the algorithms achieve.



(a) Corel5k (b) Eurlex-ev (c) Synthesized

Figure 3: Comparison of proposed distributed method with centralized methods in the literature.

We select 10, 50, 100, and 200 features on two large datasets. If there are d' features in a machine, and we want to select k of them then the runtime of the machine is $\mathcal{O}(d'k)$. Therefore, if we have $\lceil\sqrt{d/k}\rceil$ slave machines then each of them has $\mathcal{O}(\sqrt{dk})$ features and its runtime is equal to $\mathcal{O}(k\sqrt{dk})$, where d is the total number of features. Also, the master machine will have $\mathcal{O}(\sqrt{dk})$ features, and its runtime is $\mathcal{O}(k\sqrt{dk})$ which means the runtime complexity of the master machine and the slave machines are equal. If we increase or decrease the number of slave machines, then the running time of the master machine or the slave machines will increase which results in a lower speed-up. Hence, we set the number of slave machines equal to $\lceil\sqrt{d/k}\rceil$. The results show that in practice our proposed distributed algorithm achieves an approximation solution as good as the centralized algorithm in a much shorter time. The results are summarized in Table 2. Moreover, we compared the distributed and the centralized algorithms on the classification task. Results of this experiment are included in Appendix E.

Effect of λ hyper-parameter

To show the importance of both terms of the objective function, redundancy (diversity function) and relevance (submodular function), we compared the performance of the method for different λ value. We select 20, 30, 40, and 50 features on the scene dataset [Boutell et al., 2004]. As shown in Figure 2, the best performance happens for some λ between 0 and 1. This shows that both terms are necessary and it is possible to get better results by choosing λ carefully.

6 Conclusion

In this paper, we presented a greedy algorithm for maximizing the sum of a sum-sum diversity function and a non-negative, monotone, submodular function subject to a cardinality constraint in distributed and streaming settings. We showed that this algorithm guarantees a provable theoretical approximation. Moreover, we formulated the multi-label feature selection problem as such an optimization problem and developed a multi-label feature selection method for distributed and streaming settings that can handle the redundancy of the features. Improving the theoretical approximation guarantee is appealing for future work. From the empirical standpoint, it would be nice to try other metric distances and other submodular functions for the multi-label feature selection problem.

References

Abbasi Zadeh, S. and Ghadiri, M. (2015). Max-sum diversification, monotone submodular functions and semi-metric spaces. *CoRR*, abs/1511.02402.

Abbassi, Z., Mirrokni, V. S., and Thakur, M. (2013). Diversity maximization under matroid constraints. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 32–40.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2005). Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30.

Aghamolaei, S., Farhadi, M., and Zarrabi-Zadeh, H. (2015). Diversity maximization via composable coresets. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*.

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, pages 5–14.

Barbosa, R., Ene, A., Nguyen, H. L., and Ward, J. (2015). The power of randomization: Distributed submodular maximization on massive datasets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1236–1244.

Bhaskara, A., Ghadiri, M., Mirrokni, V. S., and Svensson, O. (2016). Linear relaxations for finding diverse elements in metric spaces. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4098–4106.

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2015a). Recent advances and emerging challenges of feature selection in the context of big data. *Knowl.-Based Syst.*, 86:33–45.

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2015b). Recent advances and emerging challenges of feature selection in the context of big data. *Knowl.-Based Syst.*, 86:33–45.

Borodin, A., Jain, A., Lee, H. C., and Ye, Y. (2017). Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 13(3):41:1–41:25.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771.

Chen, W., Yan, J., Zhang, B., Chen, Z., and Yang, Q. (2007). Document transformation for multi-label feature selection in text categorization. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 451–456.

Dasgupta, A., Kumar, R., and Ravi, S. (2013). Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1014–1022.

Demidova, E., Fankhauser, P., Zhou, X., and Nejdl, W. (2010). *DivQ*: diversification for keyword search over structured databases. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 331–338.

Dendamrongvit, S., Vateekul, P., and Kubat, M. (2011). Irrelevant attributes and imbalanced classes in multi-label text-categorization domains. *Intelligent Data Analysis*, 15(6):843–859.

Drosou, M. and Pitoura, E. (2010). Search result diversification. *SIGMOD Record*, 39(1):41–47.

Duygulu, P., Barnard, K., de Freitas, J. F. G., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part IV*, pages 97–112.

Francesconi, E., Montemagni, S., Peters, W., and Tiscornia, D. (2010). *Semantic processing of legal texts: Where the language of law meets the law of language*, volume 6036. Springer.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Hassin, R., Rubinstein, S., and Tamir, A. (1997). Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*

Indyk, P., Mahabadi, S., Mahdian, M., and Mirrokni, V. S. (2014). Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108.

Kashef, S., Nezamabadi-pour, H., and Nikpour, B. (2018). Multilabel feature selection: A comprehensive review and guiding experiments. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 8(2).

Khanna, R., Elenberg, E. R., Dimakis, A. G., Negahban, S., and Ghosh, J. (2017). Scalable greedy feature selection via weak submodularity. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 1560–1568.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In *Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy, April 6-8, 1994, Proceedings*, pages 171–182.

Krause, A. and Golovin, D. (2014). Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104.

Krause, A. and Guestrin, C. (2008). Beyond convexity: Submodularity in machine learning. *ICML Tutorials.*

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Liu, Z., Sun, P., and Chen, Y. (2009). Structured search result differentiation. *PVLDB*, 2(1):313–324.

Mirrokni, V. S. and Zadimoghaddam, M. (2015). Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 153–162.

Mirzasoleiman, B., Karbasi, A., Sarkar, R., and Krause, A. (2016). Distributed submodular maximization. *Journal of Machine Learning Research*, 17:238:1–238:44.

Nguyen, X. V., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854.

Peng, H., Long, F., and Ding, C. H. Q. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238.

Robnik-Sikonja, M. and Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1-2):23–69.

Spolaor, N., Cherman, E., and Monard, M. (2011). Using relieff for multi-label feature selection. In *Conferencia Latinoamericana de Informática*, pages 960–975.

Spolaôr, N., Cherman, E. A., Monard, M. C., and Lee, H. D. (2012). Filter approach feature selection methods to support multi-label learning based on relieff and information gain. In *Advances in Artificial Intelligence - SBIA 2012 - 21th Brazilian Symposium on Artificial Intelligence, Curitiba, Brazil, October 20-25, 2012. Proceedings*, pages 72–81.

Spolaôr, N., Cherman, E. A., Monard, M. C., and Lee, H. D. (2013). A comparison of multi-label feature selection methods using the problem transformation approach. *Electr. Notes Theor. Comput. Sci.*, 292:135–151.

Srivastava, A. N. and Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace conference, 2005 IEEE*, pages 3853–3862. IEEE.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, volume 21, pages 53–59. sn.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

Turnbull, D., Barrington, L., Torres, D. A., and Lanckriet, G. R. G. (2008). Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech & Language Processing*, 16(2):467–476.

Xioufis, E. S., Tsoumakas, G., and Vlahavas, I. P. (2008). An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence: Theories, Models and Applications, 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings*, pages 401–406.

Yu, C., Lakshmanan, L. V. S., and Amer-Yahia, S. (2009). It takes variety to make a world: diversification in recommender systems. In *EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings*, pages 368–378.

Zadeh, S. A., Ghadiri, M., Mirrokni, V. S., and Zadimoghaddam, M. (2017). Scalable feature selection via distributed diversity maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2876–2883.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., and Liu, H. (2010). Advancing feature selection research. *ASU feature selection repository*, pages 1–28.

A Appendix A

Proof of Lemma 2. Clearly g is non-negative and monotone. Since the sum of submodular functions is a submodular function, We only need to show that $\mathrm{top}^{\mathrm{p}}{}_{x\in S}\{MI(x,\ell)\}$ is submodular. We assume that $\mathrm{top}^{0}{}_{x\in S}\{MI(x,\ell)\} = 0$. Let $S \subseteq T \subset U$ and $a \in U \setminus T$. We show that

$$\begin{aligned}\mathop{\mathrm{top}^{\mathrm{p}}}_{x\in S\cup\{a\}}\{MI(x,\ell)\} - \mathop{\mathrm{top}^{\mathrm{p}}}_{x\in S}\{MI(x,\ell)\} \\ \geq \mathop{\mathrm{top}^{\mathrm{p}}}_{x\in T\cup\{a\}}\{MI(x,\ell)\} - \mathop{\mathrm{top}^{\mathrm{p}}}_{x\in T}\{MI(x,\ell)\}.\end{aligned}$$

We have two cases. If $MI(a,\ell)$ is not among the p largest numbers of $\{I(x,\ell)|x \in S\cup\{a\}\}$ then both sides of the above inequality are zero. If $MI(a,\ell)$ is among the p largest numbers of $\{I(x,\ell)|x \in S\cup\{a\}\}$ then the left hand side of the inequality is equal to $MI(a,\ell) - MI(b,\ell)$ where b is the p'th largest number in $\{I(x,\ell)|x \in S\}$. The right hand side is equal to $\max\{0, MI(a,\ell) - MI(c,\ell)\}$ where c is the p'th largest number in $\{I(x,\ell)|x \in T\}$. The p'th largest number in $\{I(x,\ell)|x \in T\}$ is greater than or equal to the p'th largest number in $\{I(x,\ell)|x \in S\}$ because $S \subseteq T$. Therefore, in this case $MI(a,\ell) - MI(b,\ell) \geq \max\{0, MI(a,\ell) - MI(c,\ell)\}$ and the inequality holds. □

B Appendix B

For $S \subseteq U$ and $x \in U \setminus S$, let $\Delta(x,S) = g(S\cup\{x\}) - g(S)$. We now show that Algorithm 1 is a β-nice algorithm for f. This is ultimately needed for the proof of both key lemmas.

Proof of Theorem 1. Let `ALG` be the Algorithm 1, $T \subseteq U$, $t \in T \setminus \texttt{ALG}(T)$, and $x_1, \ldots, x_k$ be the elements that `ALG` selected in the order of selection. Also, let $S_i = \{x_1, \ldots, x_i\}$ and $S_0 = \varnothing$.

For the first property of β-nice algorithms it is enough to have a consistent tiebreaking rule for ALG. It is sufficient to fix an ordering on all elements of U up front. If some iteration finds multiple elements with the same maximum marginal gain, then it should select earliest one in the a priori ordering.

Now we prove the second property of the β-nice algorithms for ALG. Because of the greedy selection of ALG, we have the following inequalities.

$$\begin{gathered}\Delta(x_1,S_0)\geq\Delta(t,S_0)\\ \Delta(x_2,S_1)+d(x_2,x_1)\geq d(t,x_1)+\Delta(t,S_1)\\ \Delta(x_3,S_2)+\sum_{i=1}^{2}d(x_3,x_i)\geq\sum_{i=1}^{2}d(t,x_i)+\Delta(t,S_2)\\ \cdots\\ \Delta(x_k,S_{k-1})+\sum_{i=1}^{k-1}d(x_k,x_i)\geq\sum_{i=1}^{k-1}d(t,x_i)+\Delta(t,S_{k-1})\end{gathered}$$

Adding these inequalities together gives the following inequality.

$$\begin{aligned}g(S_k)+D(S_k)&\geq\sum_{i=1}^{k-1}(k-i)d(t,x_i)+\sum_{i=0}^{k-1}\Delta(t,S_i)\\ &\geq\sum_{i=1}^{k-1}(k-i)d(t,x_i)+k\Delta(t,S_k),\qquad(3)\end{aligned}$$

where the second inequality holds because of the submodularity of g. Note that

$$f(\texttt{ALG}(T)\cup\{x\})-f(\texttt{ALG}(T))=\Delta(t,\texttt{ALG}(T))+\sum_{x\in\texttt{ALG}(T)}d(x,t).\qquad(4)$$

One may thus note that if the right-hand side coefficients in (3) were all $k/2$ (instead of $k-i$) we would have 2-niceness of the algorithm. Our strategy is to achieve this by shifting some of the “weight” from coefficients where $k-i>k/2$ to coefficients $<k/2$. This uses the metric inequality since $d(x_{k-i},x_i)+d(x_i,t)\geq d(x_{k-i},t)$. Hence if we added $d(x_{k-i},x_i)$ to both sides of (3), then we may increase the coefficient of $d(t,x_{k-i})$ by 1 at the expense of reducing the coefficient of $d(t,x_i)$ by 1.

We use this idea to fix all of the “small” components in bulk by adding a batch of *distinct* distances to both sides of (3). Since these distances are distinct, we increase the left-hand side by at most $D(S_k)$. In particular, the new left-hand side will be at most $2(g(S_k)+D(S_k))$.

The batch of distances we add to both sides of the inequality is $\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j)$. Clearly these distances are distinct so we now need to make sure that the strategy produces the desired coefficients of terms $d(t,x_i)$. More formally, we claim that the following inequality holds.

Claim 1.

$$\begin{aligned}\sum_{i=1}^{k-1}(k-i)d(t,x_i)+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j)\\ \geq\sum_{i=1}^{k}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)\end{aligned}$$

We prove this claim later. Using this we have the following.

$$\begin{aligned}&2(g(S_k)+D(S_k))\\ &\quad\geq g(S_k)+D(S_k)+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j)\\ &\quad\geq\sum_{i=1}^{k-1}(k-i)d(t,x_i)+k\Delta(t,S_k)\\ &\quad+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j)\\ &\quad\geq\sum_{i=1}^{k}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)+(\lceil\frac{k}{2}\rceil-1)\Delta(t,S_k)\end{aligned}$$

where the second inequalities holds because of the metric property, i.e. triangle inequality, and monotonicity of g. By using the above inequality, non-negativity of g, and (4) we have

$$\begin{aligned}\frac{2}{\lceil\frac{k}{2}\rceil-1}f(\texttt{ALG}(T))&=\frac{2}{\lceil\frac{k}{2}\rceil-1}(g(S_k)+D(S_k))\\ &\geq\sum_{i=1}^{k}d(t,x_i)+\Delta(t,S_k)\\ &=f(\texttt{ALG}(T)\cup\{t\})-f(\texttt{ALG}(T)).\end{aligned}$$

We can easily see that for $k\geq 10$, $\frac{5}{k}\geq\frac{2}{\lceil\frac{k}{2}\rceil-1}$ and $\frac{4.5}{k-1}\geq\frac{2}{\lceil\frac{k}{2}\rceil-1}$. Therefore, ALG is a 5-nice algorithm for f and because of monotonicity of g, $\frac{4.5}{k-1}f(\texttt{ALG}(T))\geq\sum_{i=1}^{k}d(t,x_i)$. □

Now we prove Claim 1 to conclude Theorem 1.

Proof of Claim 1. Note that $k=\lceil\frac{k}{2}\rceil+\lfloor\frac{k}{2}\rfloor$ and $\lfloor\frac{k}{2}\rfloor+1\geq\lceil\frac{k}{2}\rceil$. First, we show that

$$\sum_{j=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-j)d(t,x_j)=\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(t,x_j).\qquad(5)$$

In the right hand side of (5), $d(t,x_j)$ appears in the inner summation when $i-\lfloor\frac{k}{2}\rfloor-1\geq j$ or equivalently, when $i\geq j+\lfloor\frac{k}{2}\rfloor+1$. We know that $k\geq i\geq\lceil\frac{k}{2}\rceil+1$. We also know that $j\geq 1$. Hence, $j+\lfloor\frac{k}{2}\rfloor+1\geq\lceil\frac{k}{2}\rceil+1$. Therefore,

$d(t, x_j)$ definitely appears in the inner summation when $k \geq i \geq j + \lfloor \frac{k}{2} \rfloor + 1$. This means that $d(t, x_j)$ appears $k - j - \lfloor \frac{k}{2} \rfloor = \lceil \frac{k}{2} \rceil - j$ many times in the right hand side of (5). Moreover, note that the index j in the right hand side of (1) ranges between 1 and $k - \lfloor \frac{k}{2} \rfloor - 1$. Hence (5) holds. Let

$$A = \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (k-i) d(t, x_i) + \sum_{i=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i).$$

By decomposing $\sum_{i=1}^{k-1} (k-i) d(t, x_i)$ to three summations, noting that $(k-k) d(t, x_k) = 0$, and using (5), we have

$$\begin{aligned}
\sum_{i=1}^{k-1} (k-i) d(t, x_i) = & \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (k-i) d(t, x_i) \\
& + \sum_{i=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i) \\
& + \sum_{j=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (k - j - \lceil \frac{k}{2} \rceil + 1) d(t, x_j) \\
& = A + \sum_{j=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lfloor \frac{k}{2} \rfloor - j + 1) d(t, x_j) \\
& \geq A + \sum_{j=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - j) d(t, x_j) \\
& = A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} d(t, x_j).
\end{aligned}$$

Therefore, by the triangle inequality and the above statements, we have

$$\begin{aligned}
& \sum_{i=1}^{k-1} (k-i) d(t, x_i) + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} d(x_i, x_j) \\
& \geq A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} d(t, x_j) \\
& + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} d(x_i, x_j) \\
& = A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} (d(t, x_j) + d(x_i, x_j)) \\
& \geq A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor - 1} d(t, x_i) \\
& = A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} (i - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_i) \\
& \geq A + \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} (i - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_i) \\
& + (\lceil \frac{k}{2} \rceil - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_{\lceil \frac{k}{2} \rceil}) \\
& = A + \sum_{i=\lceil \frac{k}{2} \rceil}^{k} (i - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_i) \\
& = \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (k-i) d(t, x_i) + \sum_{i=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i) \\
& + \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (i - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_i) \\
& = \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (k - i + i - \lfloor \frac{k}{2} \rfloor - 1) d(t, x_i) \\
& + \sum_{i=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i) \\
& = \sum_{i=k-\lfloor \frac{k}{2} \rfloor}^{k} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i) + \sum_{i=1}^{k-\lfloor \frac{k}{2} \rfloor - 1} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i) \\
& = \sum_{i=1}^{k} (\lceil \frac{k}{2} \rceil - 1) d(t, x_i).
\end{aligned}$$

This yields the result. □

We now proceed to bound the diversity part of the optimal solution (Lemma 2). We re-use the key ideas from Aghamolaei et al. [2015] to achieve this. Let O be an optimal solution for maximizing $f(S)$ subject to $S \subseteq U$ and $|S| = k$. Let $O^i = T^i \cap O$, $Q^i = O^i \setminus S^i$. So Q^i are the elements of O on machine I that were “missed” by S^i. Intuitively, we bound the damage to optimality by missing these elements by finding a low-weight matching between Q^i and

S^i. The following normalization parameters are used in the next two lemmas: $r_i = \frac{f(S^i)}{\binom{k}{2}}$ and $r = \max_{i=1,\dots,m} r_i$. Let $G^i(O^i \cup S^i, E)$ be a complete weighted graph. For $u, v \in O^i \cup S^i$, we use $d(u,v)$ as the edge weight in our matching problem.

Lemma 4. There exists a bipartite matching between Q^i and S^i in G^i with a weight of at most $\frac{4.5}{2}|Q^i|r$ that covers all the Q^i.

Proof. The number of all maximal bipartite matchings between Q^i and S^i is $\frac{k!}{(k-|Q^i|)!}$. Any of these matchings covers Q^i because $|Q^i| \le |S^i|$. Each edge $\{q, x\}$ with $q \in Q^i$ and $x \in S^i$ is in $\frac{(k-1)!}{(k-|Q^i|)!}$ of these matchings. Hence the total weight of all matchings can be expressed as

$$\begin{aligned}\frac{(k-1)!}{(k-|Q^i|)!}\sum_{q\in Q^i}\sum_{x\in S^i} d(q,x) &\le \frac{(k-1)!}{(k-|Q^i|)!}\sum_{q\in Q^i}\frac{4.5}{k-1}f(S^i)\\ &\le \frac{(k-1)!}{(k-|Q^i|)!}\sum_{q\in Q^i}\frac{4.5}{k-1}\binom{k}{2}r\\ &= \frac{(k-1)!}{(k-|Q^i|)!}|Q^i|\frac{4.5k}{2}r\\ &= \frac{k!}{(k-|Q^i|)!}\frac{4.5}{2}|Q^i|r\end{aligned}$$

The first inequality is from Lemma 1 and the second by the definition of r. It follows that there exists a matching with a weight of at most $\frac{4.5}{2}|Q^i|r$. □

We are now in position to upper bound the diversity portion of an optimal solution in terms of $f(\texttt{OPT}(\cup_i^m S^i))$.

Proof of Lemma 2. Let M^i be the maximal bipartite matching between Q^i and S^i with a weight of less than or equal to $\frac{4.5}{2}|Q^i|r$. It exists because of Lemma 4. Let $M = \cup_{i=1}^m M^i$. Note that S_i's are disjoint and Q^i's are disjoint. This implies that M^i's are disjoint. Therefore, M is a matching between $\cup_{i=1}^m Q^i$ and $\cup_{i=1}^m S^i$ that covers all of $\cup_{i=1}^m Q^i$ with a weight of less than or equal to $\frac{4.5}{2}\sum_{i=1}^m |Q^i|r \le \frac{4.5}{2}|O|r = \frac{4.5}{2}kr$.

Let $e : O \to \cup_{i=1}^m S^i$ be a mapping which maps any $o \in O \cap (\cup_{i=1}^m S^i)$ to itself and any $o \in (\cup_{i=1}^m Q^i)$ to its matched vertex in M. The weight of this mapping is less than or equal to the weight of M since $d(o,o) = 0$. Note that each vertex in the $range(e)$ is mapped from at most two vertices in O. We use this fact in the second inequality below and use the triangle inequality in the first inequality. We have

$$\begin{aligned}D(O) &= \sum_{\{u,v\}\in O} d(u,v)\\ &\le \sum_{\{u,v\}\in O}(d(u,e(u)) + d(e(u),e(v)) + d(e(v),v))\\ &= (|O|-1)\sum_{u\in O} d(o,e(o)) + \sum_{\{u,v\}\in O} d(e(u),e(v))\\ &\le (k-1)\frac{4.5}{2}kr + 4D(range(e))\\ &\le 4.5\binom{k}{2}r + 4f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &\le 8.5f(\texttt{OPT}(\cup_{i=1}^m S^i))\end{aligned}$$

□

Now, we proceed to bound $g(O)$ and the proofs of the next two lemmas follow those found in Mirrokni and Zadimoghaddam [2015]. Let $o_1, \dots, o_k$ be an ordering of elements of O. For $x = o_i \in O$ define $O_x = \{o_1, \dots, o_{i-1}\}$ and $O_{o_1} = \varnothing$.

Lemma 5. $g(O) \le 6f(\texttt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x, O_x \cup S^i))$.

Proof. Note that $g(O) = g(O \cap (\cup_{i=1}^m S^i)) + \sum_{x\in O\setminus(\cup_{i=1}^m S^i)} \Delta(x, O_x \cup (O \cap (\cup_{i=1}^m S^i)))$. Therefore, using submodularity and monotonicity of g and 5-niceness of Algorithm 1, we have

$$\begin{aligned}g(O) &\le f(\texttt{OPT}(\cup_{i=1}^m S^i)) + \sum_{x\in O\setminus(\cup_{i=1}^m S^i)}\Delta(x,O_x)\\ &= f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &+ \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x\cup S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i))\\ &\le f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &+ \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i))\\ &\le f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &+ \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\frac{5}{k}f(S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i))\\ &\le f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &+ \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\frac{5}{k}f(\texttt{OPT}(\cup_{i=1}^m S^i)) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i))\\ &\le f(\texttt{OPT}(\cup_{i=1}^m S^i)) + 5f(\texttt{OPT}(\cup_{i=1}^m S^i))\\ &+ \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x\cup S^i))\\ &\le 6f(\texttt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x\cup S^i))\end{aligned}$$

□

In the next Lemma, we use the randomness of the partitioning of the data over machines and the first property of β-niceness.

Lemma 6. $\mathbb{E}[\sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x \cup S^i))] \le \mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^m S^i))]$.

Proof. We show that $\mathbb{E}[\sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x \cup S^i))] \le \frac{\mathbb{E}[\sum_{i=1}^m g(S^i)]}{m}$ and the statement of the lemma follows from the fact that $\frac{\sum_{i=1}^m g(S^i)}{m} \le f(\texttt{OPT}(\cup_{i=1}^m S^i))$. We first establish an inequality

$$A := \mathbb{E}[\sum_{i=1}^m\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x\cup S^i))] \le \frac{1}{m}B$$

where

$$B := \mathbb{E}[\sum_{i=1}^m\sum_{x\in O}(\Delta(x,O_x) - \Delta(x,O_x\cup S^i))].$$

Let $\texttt{ALG}$ be Algorithm 1. For $T \subseteq U$ and $x \in U$, let $q(x,T) = \Delta(x, O_x) - \Delta(x, O_x \cup \texttt{ALG}(T))$. Let $P[.]$ be the probability mass function for the uniform distribution over m-partitions $\mathbb{P} = (T^1, \dots, T^m)$ of U, and let $\mathbb{1}[x \notin \texttt{ALG}(T \cup \{x\})]$ be a $0, 1$ indicator function. Note that

$$P[T^i = T] = (\frac{1}{m})^{|T|}(1 - \frac{1}{m})^{|U|-|T|}$$

$$P[T^i = T \cup \{x\}] = (\frac{1}{m})^{|T|+1}(1 - \frac{1}{m})^{|U|-|T|-1}$$

Therefore

$$P[T^i = T \cup \{x\}] = \frac{P[T^i = T] + P[T^i = T \cup \{x\}]}{m}. \quad (6)$$

We have that

$$A = \sum_{i=1}^{m} \sum_{x \in O} \sum_{T \subseteq U \setminus \{x\}} P[T^i = T \cup \{x\}] \mathbb{1}[x \notin \texttt{ALG}(T \cup \{x\})] q(x, T \cup \{x\})$$

$$B = \sum_{i=1}^{m} \sum_{x \in O} \sum_{T \subseteq U \setminus \{x\}} (P[T^i = T \cup \{x\}] q(x, T \cup \{x\}) + P[T^i = T] q(x, T))$$

$$\geq \sum_{i=1}^{m} \sum_{x \in O} \sum_{T \subseteq U \setminus \{x\}} \mathbb{1}[x \notin \texttt{ALG}(T \cup \{x\})] q(x, T \cup \{x\}) (P[T^i = T \cup \{x\}] + P[T^i = T]).$$

The last inequality holds because $q(.,.)$ is a non-negative function and multiplying it by $\mathbb{1}[x \notin \texttt{ALG}(T \cup \{x\})]$ can only decrease the sum value. Also, $q(x,T)$ is replaced by $q(x, T \cup \{x\})$. It does not change the sum value because when $\mathbb{1}[x \notin \texttt{ALG}(T \cup \{x\})] = 1$, $q(x,T) = q(x, T \cup \{x\})$.

We now deduce $A \leq B/m$ from (6).

Now note that $\sum_{x \in O} \Delta(x, O_x \cup S^i) = g(O \cup S^i) - g(S^i)$, and $\sum_{x \in O} \Delta(x, O_x) = g(O)$. Therefore, because of the monotonicity of g, we have for any i

$$\sum_{x \in O} \Delta(x, O_x) - \Delta(x, O_x \cup S^i)$$
$$= g(O) - g(O \cup S^i) + g(S^i) \leq g(S^i).$$

Hence $B \leq \frac{\mathbb{E}[\sum_{i=1}^{m} g(S^i)]}{m}$ and the lemma follows. □

We now have that Lemma 3 follows directly from Lemmas 5, and 6 as they imply

$$g(O) \leq 6f(\texttt{OPT}(\cup_{i=1}^{m} S^i)) + \mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^{m} S^i))].$$

Therefore this completes the proof of Theorem 2.

C Appendix C

Let n be the number of samples in the dataset, L_i be the set of labels for sample i that are 1 in the dataset, and L'_i be the set of labels for sample i that we predicted to be 1. Then the subset accuracy of our learning method is equal to

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(L_i, L'_i)$$

where $\mathbb{I}(,.,)$ is a 0, 1 indicator function and is equal to 1 when set L_i is equal to the set L'_i, and it is 0 otherwise. Example-based accuracy is equal to

$$\frac{1}{n} \sum_{i=1}^{n} \frac{|L_i \cap L'_i|}{|L_i \cup L'_i|}.$$

Example-based F-measure is equal to

$$\frac{1}{n} \sum_{i=1}^{n} \frac{2|L_i \cap L'_i|}{|L_i| + |L'_i|}.$$

These evaluation measures are example-based. Micro-averaged F-measure and Macro-averaged F-measure are two label-based measures for multi-label classification. Let t be the number of labels in the dataset, E_i be the set of examples that their i'th label is equal to 1, and E'_i be the set of example that we predicted their i'th labels to be 1. Then Micro-averaged F-measure is equal to

$$\frac{1}{t} \sum_{i=1}^{t} \frac{2|E_i \cap E'_i|}{|E_i| + |E'_i|}.$$

Macro-averaged F-measure is equal to

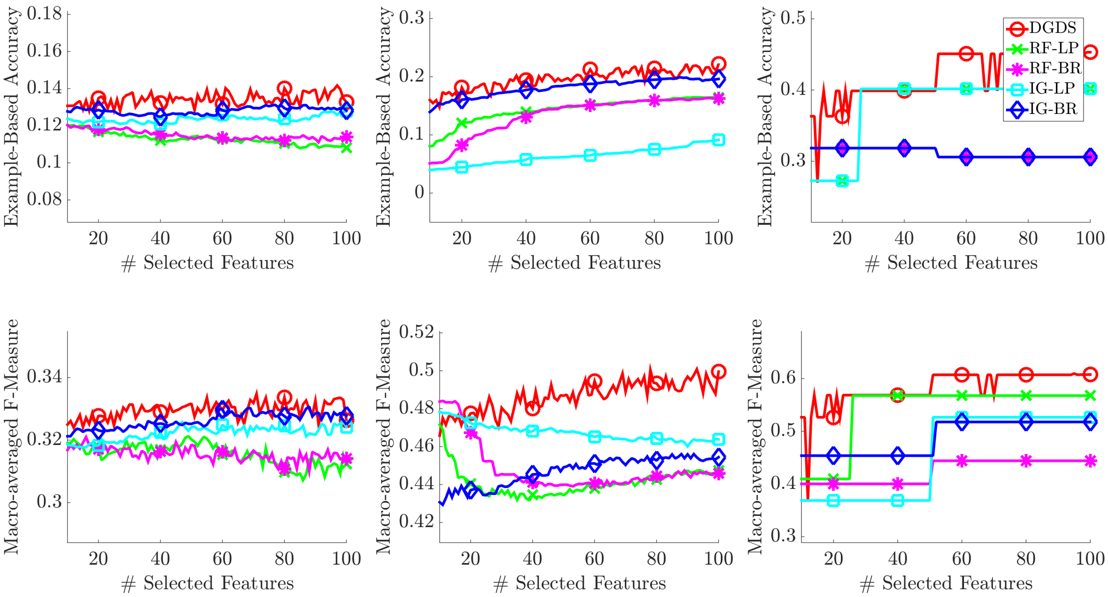
$$\frac{2 \sum_{i=1}^{t} |E_i \cap E'_i|}{\sum_{i=1}^{t} |E_i| + \sum_{i=1}^{t} |E'_i|}.$$

D Appendix D

Table 3: Specifications of other datasets.

Dataset Name	# Features	# Instances	# Labels	Reference
CAL500	68	502	174	[Turnbull et al., 2008]
Delicious	500	16,105	983	[Tsoumakas et al., 2008]
Scene	294	2407	6	[Boutell et al., 2004]

E Appendix E

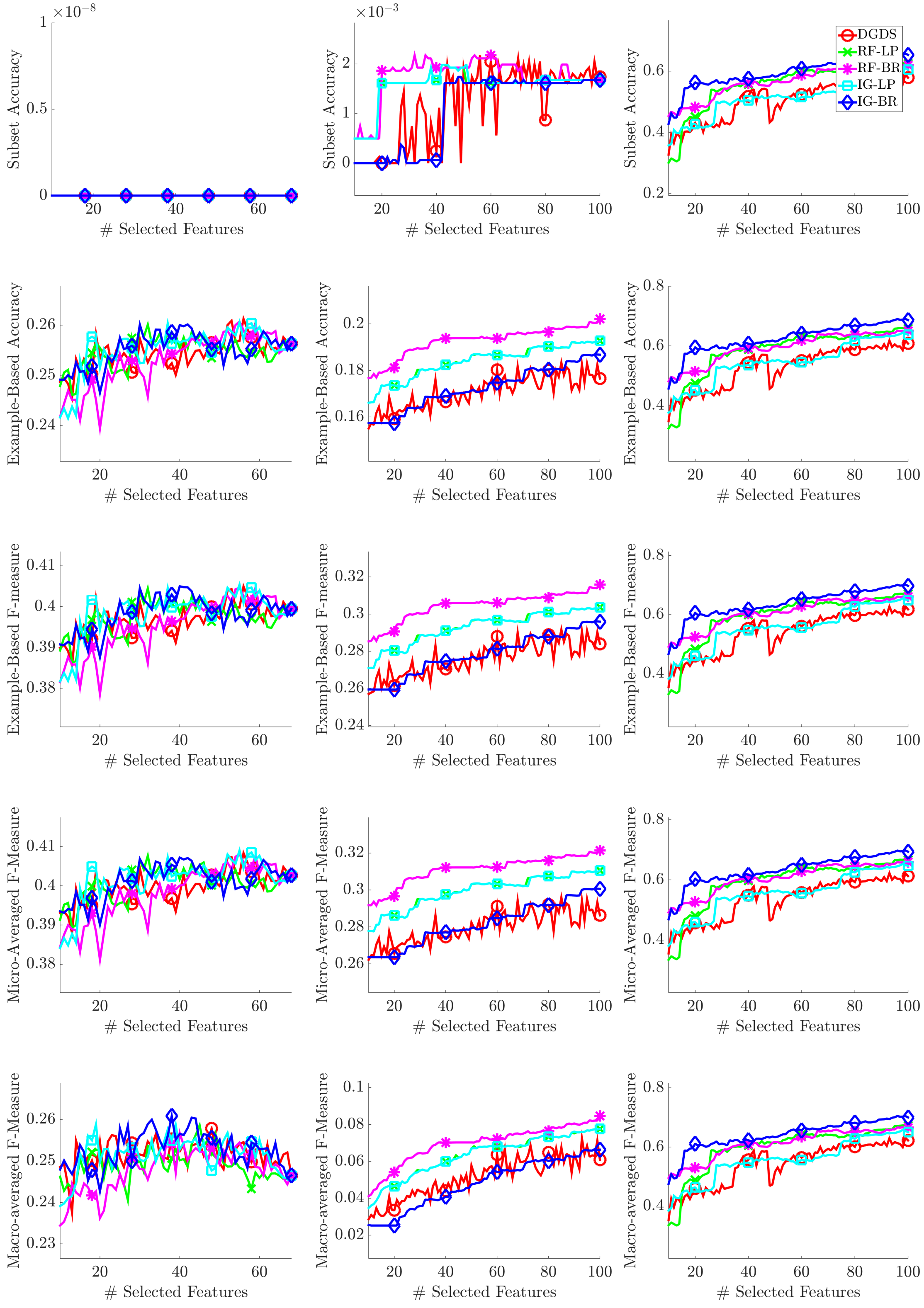


(a) Corel5k

(b) Eurlex-ev

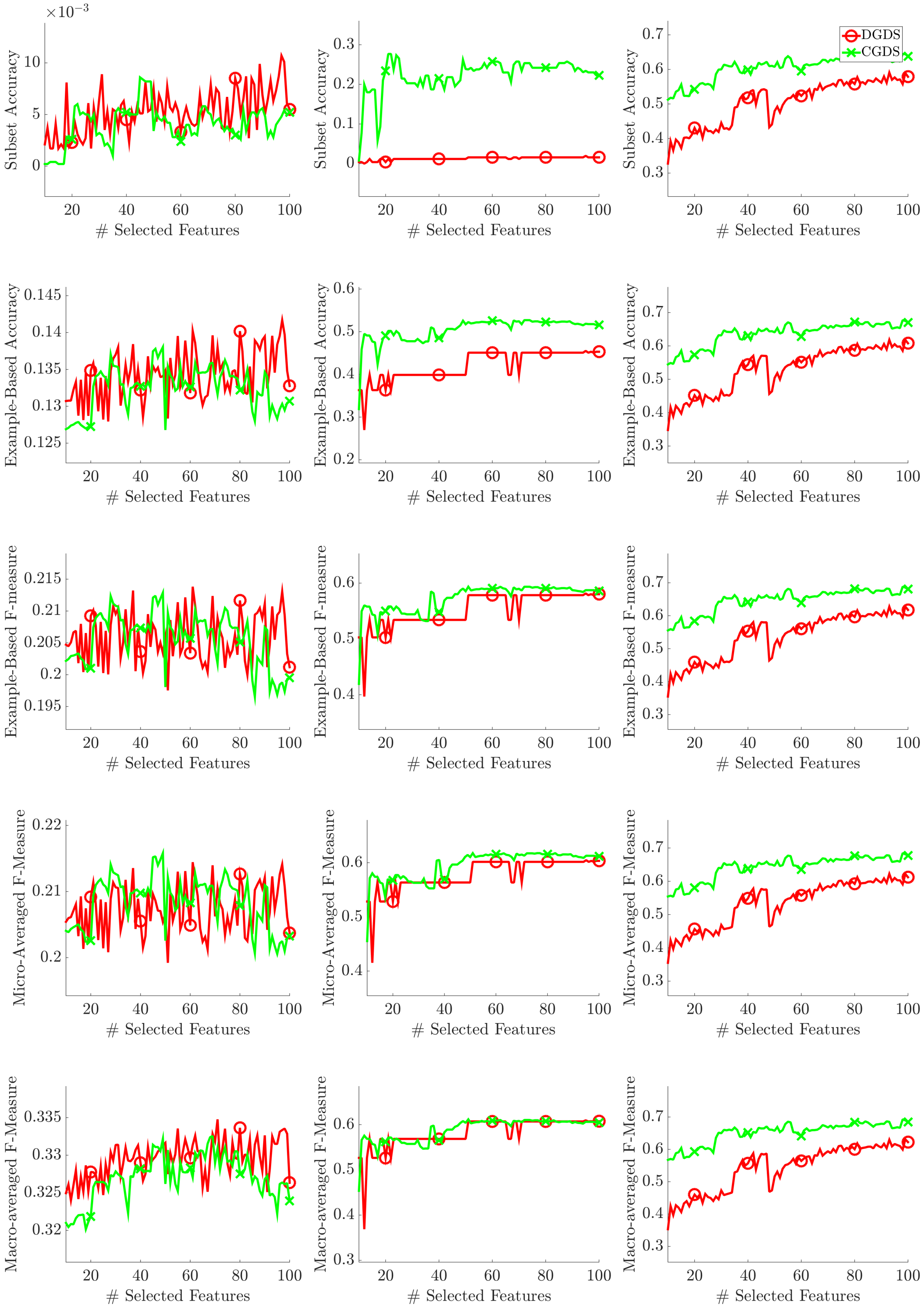
(c) Synthesized

Figure 4: Comparison of proposed distributed method with centralized methods in the literature.



(a) CAL500 (b) Delicious (c) Scene

Figure 5: Comparison of proposed distributed method with centralized methods in the literature.



(a) Corel5k (b) Synthesized (c) Scene

Figure 6: Comparison of proposed distributed method (DGDS) with proposed centralized method (CGDS) on the classification task.