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

Mehrdad Ghadiri  
University of British Columbia

Mark Schmidt  

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, Drossou 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 streaming 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 $\sqrt{d/k}$ 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-
selected 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 [Kashif 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 \( \ell_0 \) and \( \ell_1 \) 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 methods.

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, streaming, 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]. A 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 \( A \subseteq B \subset U \). 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 \( f(S) + \lambda \sum_{\{u,v\} \mid u,v \in S} d(u,v) \) 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 \( U \), \( (T^1, \ldots, T^m) \) be a random partitioning of elements of \( U \), and \( k \) be a positive integer. Let \( \text{OPT}(T) = \arg \max_{S \subseteq T} \{ k \cdot f(S) \mid T \subseteq U \} \). Let \( \text{ALG} \) be an algorithm which takes \( T \subseteq U \) as an input and outputs \( S \subseteq T \). For \( \alpha > 0 \), we call \( \text{ALG} \) an \( \alpha \)-approximate composable core-set with size \( k \) for \( f \) if the size of its output is \( k \) and \( f(\text{ALG}(T^1 \cup \cdots \cup \text{ALG}(T^m))) \geq \alpha f(\text{OPT}(T^1 \cup \cdots \cup T^m)) \) [Indyk et al., 2014]. We call \( \text{ALG} \) an \( \alpha \)-approximate randomized composable core-set with size \( k \) for \( f \) if the size of its output is \( k \) and \( E[f(\text{ALG}(T^1 \cup \cdots \cup \text{ALG}(T^m)))] \geq \alpha f(\text{OPT}(T^1 \cup \cdots \cup T^m)) \) [Borodin et al., 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 \( \frac{1}{2} \) 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 \( \frac{1}{2} \) 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 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 $I(\ldots)$ 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 [Kashid 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

$$d(u_i, u_j) = 1 - \frac{I(u_i, u_j)}{H(u_i, u_j)}$$

$$= 1 - \frac{\sum_{x \in u_i, y \in u_j} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}}{-\sum_{x \in u_i, y \in u_j} p(x, y) \log p(x, y)},$$

where $H(\ldots)$ is the joint entropy and $I(\ldots)$ 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 $g : 2^L \rightarrow \mathbb{R}$ which measures the relevance of the selected features to the labels. For any positive integer $p$, we define

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

where $\top^p \{MI(x, \ell)\}$ 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 $H(\cdot)$ is the entropy function and the value $MI(\cdot, \cdot)$
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:

\[
\max_{S \subseteq U \atop |S| = k} f(S) = \max_{S \subseteq U \atop |S| = k} \left( g(S) + \sum_{(u,v) \in S} d(u, v) \right),
\]

(1)

where \( d(\cdot, \cdot) \) is a metric distance and \( g(\cdot) \) 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**

1. **Input:** Set of features \( U \), set of labels \( L \), number of features we want to select \( k \).
2. **Output:** Set \( S \subseteq U \) with \( |S| = k \).
3. \( S \leftarrow \{ \text{arg max}_{u \in U} g(\{u\}) \} \).
4. **forall** 2 \( \leq i \leq k \) do
   5. \( u^* \leftarrow \text{arg max}_{u \in U \setminus S} g(S \cup \{u\}) - g(S) + \sum_{x \in S} d(x, u) \);
   6. Add \( u^* \) to \( S \);
5. 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 \( \frac{1}{k} \). 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 4. 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, \( \lambda \) is a hyper-parameter which controls the effect of two criteria on the final function.

**Algorithm 2: AltGreedy**

1. **Input:** Set of features \( U \), set of labels \( L \), number of features we want to select \( k \).
2. **Output:** Set \( S \subseteq U \) with \( |S| = k \).
3. \( S \leftarrow \{ \text{arg max}_{u \in U} g(\{u\}) \} \);
4. **forall** \( 2 \leq i \leq k \) do
   5. \( u^* \leftarrow \text{arg max}_{u \in U \setminus S} \frac{1}{2} g(S \cup \{u\}) - g(S) + \sum_{x \in S} d(x, u) \);
   6. Add \( u^* \) to \( S \);
5. 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 \( d(\cdot, \cdot) \) is a metric distance. In this section, we show that Algorithm 2 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 2 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**

1. **Input:** Set of features \( U \), set of labels \( L \), number of features we want to select \( k \), number of machines \( m \).
2. **Output:** Set \( S \subseteq U \) with \( |S| = k \).
3. Randomly partition \( U \) into \( (T_i)_{i=1}^m \);
4. **forall** \( 1 \leq i \leq m \) do
   5. \( S_i \leftarrow \text{output of Greedy}(T_i, L, k) \);
   6. \( S \leftarrow \text{output of AltGreedy}(\cup_{i=1}^m S_i, L, k) \);
5. Return \( S \);

We use the following key concept of a \( \beta \)-nice algorithm from [Mirrokni and Zadimoghaddam, 2015] throughout our analysis.
We note that for Algorithm 1, we focus on the large $O$ in Appendix B. Theorem 1. Algorithm 1 is a half approximation algorithm for maximizing $f$. Therefore, if ALG is Algorithm 1 then $f(OPT(U^{m}_{i=1}S^{i})) \leq 2f(ALG^*(U^{m}_{i=1}S^{i}))$. Hence $f(O) \leq 31E[f(ALG^*(U^{m}_{i=1}S^{i}))]$ 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 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.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Reference</th>
<th># Features</th>
<th># Instances</th>
<th># Labels</th>
<th># Selected Features</th>
<th># Machines</th>
<th>Distributed Algorithm Objective Value</th>
<th>Centralized Algorithm Objective Value</th>
<th>Distributed Algorithm Runtime</th>
<th>Centralized Algorithm Runtime</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCV1V2</td>
<td>Lewis et al. 2004</td>
<td>47,236</td>
<td>6000</td>
<td>101</td>
<td>10</td>
<td>69</td>
<td>22.7</td>
<td>22.6</td>
<td>2.8m</td>
<td>1h 33m</td>
<td>31.2</td>
</tr>
<tr>
<td>TMC2007</td>
<td>Srivastava and Zane-Ulman 2007</td>
<td>49,600</td>
<td>28,596</td>
<td>22</td>
<td>50</td>
<td>71</td>
<td>22.8</td>
<td>22.6</td>
<td>4.6m</td>
<td>2h 32.5m</td>
<td>33.4</td>
</tr>
</tbody>
</table>

**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 Spolaor et al. 2012 2013.

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 5) 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.
Multi-Label Feature Selection Using “Submodular Plus Diversity” Maximization

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 \( O(d'k) \). Therefore, if we have \( \lceil \sqrt{d/k} \rceil \) slave machines then each of them has \( O(\sqrt{dk}) \) features and its runtime is equal to \( O(k\sqrt{dk}) \), where \( d \) is the total number of features. Also, the master machine will have \( O(\sqrt{dk}) \) features, and its runtime is \( O(k\sqrt{dk}) \) 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 \( \lambda \) 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 \( \lambda \) 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 \( \lambda \) between 0 and 1. This shows that both terms are necessary and it is possible to get better results by choosing \( \lambda \) 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


