Scalable Feature Selection via Distributed Diversity Maximization

Sepehr Abbasi Zadeh*, Mehrdad Ghadiri*
Department of Computer Engineering
Sharif University of Technology
{sabbasizadeh, ghadiri}@ce.sharif.edu

Vahab Mirrokni, Morteza Zadimoghaddam
Google Research
{mirrokni, zadim}@google.com

Abstract

Feature selection is a fundamental problem in machine learning and data mining. The majority of feature selection algorithms are designed for running on a single machine (centralized setting) and they are less applicable to very large datasets. Although there are some distributed methods to tackle this problem, most of them are distributing the data horizontally which are not suitable for datasets with a large number of features and few number of instances. Thus, in this paper, we introduce a novel vertically distributable feature selection method in order to speed up this process and be able to handle very large datasets in a scalable manner. In general, feature selection methods aim at selecting relevant and non-redundant features (Minimum Redundancy and Maximum Relevance). It is much harder to consider redundancy in a vertically distributed setting than a centralized setting since there is no global access to the whole data. To the best of our knowledge, this is the first attempt toward solving the feature selection problem with a vertically distributed filter method which handles the redundancy with consistently comparable results with centralized methods. In this paper, we formalize the feature selection problem as a diversity maximization problem by introducing a mutual-information-based metric distance on the features. We show the effectiveness of our method by performing an extensive empirical study. In particular, we show that our distributed method outperforms state-of-the-art centralized feature selection algorithms on a variety of datasets. From a theoretical point of view, we have proved that the used greedy algorithm in our method achieves an approximation factor of $1/4$ for the diversity maximization problem in a distributed setting with high probability. Furthermore, we improve this to $8/25$ expected approximation using multiplicity in our distribution.

Introduction

Feature selection is the task of choosing a small representative subset of features from a dataset. It is a crucial preprocessing step in several data mining and machine learning applications as it can significantly reduce the learning time and the error rate (Guyon and Elisseeff 2003). This problem has received even more attention in the big data era with ultrahigh-dimensional datasets in different fields such as bioinformatics (Greene et al. 2014), neuroscience (Turk-Browne 2013), finance, and document classification. Although there are various methods that are fast enough to handle several thousands of features, they become less applicable in the presence of millions or billions of features. To deal with such big datasets, we need more scalable and faster approaches such as distributed methods. Most of the developed distributed methods distribute the data by instances among machines—*horizontally distributing*—(Zhao et al. 2012; Peralta et al. 2015) and there are a few recent works tackling the problem by distributing the feature vectors—*vertically distributing* (Moran-Fernandez, Bolón-Canedo, and Alonso-Betanzos 2015). Intuitively, horizontal distribution is infeasible when the dataset has many features but a few instances, e.g., bioinformatics datasets. As a matter of fact, it is harder to deal with the redundancy of the features in a vertically distributed setting in comparison to horizontally distributed or centralized settings (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015b). In the rest of the paper we use *centralized* setting in contrast to *distributed* setting.

Over the past decade, several information-theoretic methods have been developed for feature selection. In addition to having a strong theoretical foundation, these techniques have shown reasonable performance in practice. They consider two intuitive criteria: minimizing redundancy of the selected features by reducing mutual information between them, and maximizing their relevance by increasing mutual information between them and class labels. These criteria are the foundations of a widely used and well-known feature selection framework called mRMR (Ding and Peng 2005). In this paper, we note that a potential way of considering these criteria in a distributed setting is through diversity maximization framework. Therefore, our method can be viewed as a distributed version of mRMR that also guarantees a theoretical bound for its objective function.

In the diversity maximization framework, the goal is to find a fixed-sized set of points from a metric space which maximizes a certain diversity function. This diversity function is defined based on a distance between elements of the metric space (Abbassi, Mirrokni, and Thakur 2013; Indyk et al. 2014; Abbasi Zadeh and Ghadiri 2015). As one of our main contributions toward modeling feature selection through diversity maximization, we introduce a metric distance between feature vectors which considers their redun-

*Both authors contributed equally to this work dancy and relevance, simultaneously.

Recently, several techniques have been developed to solve diversity maximization problems in a distributed manner (Indyk et al. 2014; Aghamolaei, Farhadi, and Zarrabi-Zadeh 2015). However, these algorithms can be improved in terms of provable approximation factors and also in terms of the computational complexity of the underlying algorithm.

Our Contributions

Our contributions in this paper are three-fold:

- We introduce a new model for filter-based feature selection by formalizing it as a diversity maximization problem defined on a metric distance function among features. More specifically, the objective function is a linear combination of the sum of pairwise distances of features and their mutual information with class labels.
- We present an improved constant-factor approximate distributed algorithm for diversity maximization which is also faster than the existing algorithms. This algorithm is a distributed version of the well-studied greedy algorithm for this problem. In order to achieve the approximation factor of $\frac{8}{25}$, which is a great improvement over the latest existing approximation factor of $\frac{1}{12}$, we apply two algorithmic techniques: 1) By random partitioning of the features, we obtain an approximately optimal *randomized composable core-set* for this problem (defined by (Mirrokni and Zadimoghaddam 2015)), and 2) To improve approximation factor from $\frac{1}{4}$ to $\frac{8}{25} - \epsilon$, instead of sending each feature to a single machine, we send it to $1/\epsilon$ randomly chosen machines.
- We perform an extensive empirical study on a variety of dataset types including biological, texts, and images covering a broad range of number of features from a few thousands to millions, and show the effectiveness of our filter-based distributed algorithm from three perspectives: 1) By simulating a distributed system on a single machine, we show that our distributed method outperforms the existing centralized state-of-the-art methods in terms of classification accuracy, while we also significantly improve the running time. To do so, we compare with various feature selection algorithms. 2) We show the advantages of the greedy over the local search method for the distributed diversity maximization. 3) Finally, we demonstrate our objective function's quality by considering all the fixed-sized combinations of features on small-sized datasets and show a high correlation between the diversity objective value and the classification accuracy.

Related Work

Diversity Maximization. The diversity maximization problems generalize the maximum dispersion problem (Hassin, Rubinstein, and Tamir 1997). This problem has been explored in the context of diversity maximization for recommender systems, and commerce search. A number of $\frac{1}{2}$-approximation algorithms have been developed for the centralized version of this problem and its generalizations (Hassin, Rubinstein, and Tamir 1997; Abbassi, Mirrokni, and Thakur 2013; Borodin, Lee, and Ye 2014). More recently, it has been independently shown by (Bhaskara et al. 2016) and (Borodin, Lee, and Ye 2014) that under the planted clique conjecture, improving this $\frac{1}{2}$-approximation for the diversity maximization problem is not possible.

Composable Core-sets. Recently, a new concept has emerged in distributed computing literature. An α-approximate composable core-set for an optimization problem like P is a mapping from a set to a subset of it, so that the union of the subsets for a collection of sets includes a solution within α factor of the optimal solution of the union of the sets. Several diversity maximization problems in a metric space have been examined using composable core-sets (Indyk et al. 2014). In that work, a constant-factor approximation has been given for the distributed diversity maximization problem examined in this paper, however, the approximation factor of the algorithm is a large constant ($< \frac{1}{100}$). This factor has been improved to $\frac{1}{12}$ in (Aghamolaei, Farhadi, and Zarrabi-Zadeh 2015). There is still a large gap between the $\frac{1}{12}$ approximation factor and the potential $\frac{1}{2}$ approximation guarantee (Birnbaum and Goldman 2009). Here, we present a much improved approximation algorithm and tighten this gap. The idea of using random partitioning and multiplicity in distributed optimization have been applied to submodular maximization (Mirrokni and Zadimoghaddam 2015). Note that the diversity function is not submodular, and the proof techniques used for submodular maximization may not be applied to the diversity maximization.

Feature Selection. Feature selection methods are divided into three general families: filters, wrappers and embedded methods. Filter methods are independent from the learning algorithm that will be used on the data. On the contrary, wrapper methods apply a learning algorithm in order to evaluate the feature subsets according to their performance and embedded methods select the features during the training process. In general, filter methods are faster than the others and more capable to prevent overfitting (Guyon and Elisseeff 2003). Mutual-information-based methods are a well-known family of filter methods. They attempt to find non-redundant features which are relevant to the class labels (Peng, Long, and Ding 2005).

There have been a great amount of progression on the centralized feature selection methods in the last decades, in which all the data had to be stored and processed on a single machine. Although the proposed centralized methods are quite fast and efficient, in the context of big data, they cannot perform well and the need for distributed methods are crucial. The data can be partitioned in two ways among machines: horizontally and vertically. We refer to distributing the data by instances as *horizontal partitioning* and distributing the data by features as *vertical partitioning*. Horizontal partitioning cannot handle datasets with few number of instances and large number of features. Although there were some attempts for vertically distributed feature selection methods, most of them do not consider the redundancy of the features (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015b). In (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015a) and (Sharma, Imoto, and Miyano

2012), some wrapper methods were introduced in order to overcome the redundancy issue by using the classification accuracy. To the best of our knowledge, prior to our work, there have been no vertically distributed filter feature selection method that can deal properly with the redundancy.

Modeling Feature Selection with Diversity Maximization

Let $\mathbb{U}$ be a set of size d of labeled instances accompanied by a large set of features $\mathbb{N}$ with cardinality n, represented in a $d \times n$ matrix. The instances are labeled with a d-dimensional vector L known as the class label vector. In the feature selection problem, we aim to find a compact subset of features S that explains the labels with high accuracy. To do so, we would like to select a diversified subset of features which are relevant to the vector of class labels. Hence, if we define a metric distance between features that considers redundancy and relevance, the feature selection problem will reduce to the diversity maximization problem.

Definition 1. *For two feature vectors like p and q and the class label vector L, we define the $\textsc{Dist}(p,q)$ as follows:*

$$\textsc{Dist}(p,q) = \begin{cases} \lambda \mathrm{VI}(p,q) + (1-\lambda)\frac{(\mathrm{MI}(p,L)+\mathrm{MI}(q,L))}{2} & p \neq q \\ 0 & p = q \end{cases} \tag{1}$$

where MI is the normalized mutual information and VI is the normalized variation of information. The first term of this definition is for avoiding the redundancy. The second term is for maximizing the relevance and λ is a regularization factor which determines the importance of each criterion. The normalized mutual information of two discrete random variables X and Y is defined as follows:

$$\begin{aligned} \mathrm{MI}(X,Y) &= \frac{I(X;Y)}{\sqrt{H(X)H(Y)}} \\ &= \frac{\sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}}{\sqrt{\sum_{x\in X}(p(x)\log p(x))\sum_{y\in Y}(p(y)\log p(y))}}, \end{aligned} \tag{2}$$

where $I(.;.)$ is the mutual information function, $H(.)$ is the entropy function and $p(.,.)$ is the joint probability. Also, the normalized variation of information of two discrete random variables X and Y is defined as follows:

$$\begin{aligned} \mathrm{VI}(X,Y) &= 1 - \frac{I(X;Y)}{H(X,Y)} \\ &= 1 - \frac{\sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}}{-\sum_{x\in X, y\in Y} p(x,y)\log p(x,y)}, \end{aligned} \tag{3}$$

where $H(.,.)$ is the joint entropy. To compute the presented distance for feature vectors with continuous value, we discretize them. We elaborate more in the experiments section.

To prove that the $\textsc{Dist}$ is metric, i.e., $\textsc{Dist}(p,q) + \textsc{Dist}(q,r) \geq \textsc{Dist}(p,r)$, we point that VI is metric (Nguyen, Epps, and Bailey 2010) and the mutual information terms will unfold as follows: $\mathrm{MI}(p,L)+\mathrm{MI}(q,L)+\mathrm{MI}(q,L)+\mathrm{MI}(r,L) \geq \mathrm{MI}(p,L)+\mathrm{MI}(r,L)$ which holds because of the non-negativity of $\mathrm{MI}(.,L)$.

We want to maximize the following objective function (as the objective of diversity maximization problem) for $S \subset \mathbb{N}$ and $|S| = k$.

$$\begin{aligned} \textsc{Div}(S) &= \frac{1}{2}\sum_{p\in S}\sum_{q\in S}\textsc{Dist}(p,q) \\ &= \lambda\frac{1}{2}\sum_{p\in S}\sum_{q\in S}\mathrm{VI}(p,q) + (1-\lambda)\frac{(k-1)}{2}\sum_{p\in S}\mathrm{MI}(p,L) \end{aligned} \tag{4}$$

The first term of Equation 4 prevents the selection of redundant features because the VI of two similar features is near to zero and the second term makes sure that the selected features be relevant to the class labels. At this point we have modeled the feature selection problem through diversity maximization (Equation 4). Hence, in the next section we focus on finding a set of k points S with maximum diversity $\textsc{Div}(S)$ in a distributed setting. In our analysis, *feature* and *point* are being used interchangeably.

Algorithm DDisMI

In this section we describe the theoretical foundations of our distributed method. Before that, it is worth mentioning that our metric distance and objective function can be used in a centralized algorithm similar to mRMR. Considering the result of (Birnbaum and Goldman 2009), in which it is shown that the centralized greedy selection of the points in a metric space using Algorithm 1 achieves a $\frac{1}{2}$ approximation factor for diversity maximization problem, our method, in its centralized setting, guarantees a $\frac{1}{2}$ approximation factor for its objective as well, whereas the mRMR method does not provide such a theoretical guarantee. Hereafter, we propose the distributed greedy-based algorithm DDisMI, and prove that it achieves an approximation factor of $\frac{1}{4} - \epsilon$ for a small sub-constant $\epsilon = o(1)$. To get further improvements, we slightly change the distribution of points among machines, and achieve an approximation factor of $\frac{8}{25} - \epsilon$.

Algorithm 1: Greedy

```
Input: Set of points $T$, $k$.
Output: Set $S \subset T$ with $|S| \leq k$.
$S \leftarrow \{$an arbitrary point $p \in T\}$;
forall $2 \leq i \leq k$ do
    $p^* \leftarrow \arg\max_{p\in T\setminus S}\sum_{q\in S}\textsc{Dist}(p,q)$;
    Add $p^*$ to $S$;
Return $S$;
```

Algorithm 2 (DDisMI) consists of two main phases. In phase one, corresponding to lines $3 - 6$, we partition all points randomly into m parts $\{T_\ell\}_{\ell=1}^{m}$, and give set T_ℓ to a machine ℓ for each $1 \leq \ell \leq m$ where m is the number of machines. Machine ℓ runs Algorithm 1 (Greedy) on T_ℓ, and selects a set S_ℓ of k points. In the second and final phase, we put all the selected sets together, and run the Greedy algorithm again on their union to achieve set $S \subset \cup_{\ell=1}^{m} S_\ell$ of k points. Among this set S, and the m selected sets $\{S_\ell\}_{\ell=1}^{m}$, we output the set with the maximum diversity. In practice, it suffices to output the set S and this extra comparison (lines

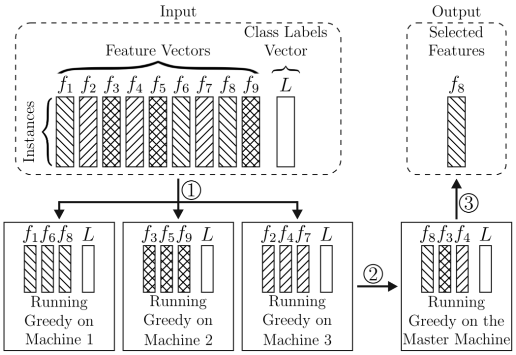

Input
Feature Vectors
Class Labels Vector
f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9
L
Instances
Output
Selected Features
f_8
①
②
③
$f_1 f_6 f_8$ L
Running Greedy on Machine 1
$f_3 f_5 f_9$ L
Running Greedy on Machine 2
$f_2 f_4 f_7$ L
Running Greedy on Machine 3
$f_8 f_3 f_4$ L
Running Greedy on the Master Machine

Figure 1: Illustration of the used distributed setting in the DDisMI algorithm. First, feature vectors are distributed randomly between m machines (Step 1), then each machine selects k features using the Greedy algorithm and sends them to the master machine (Step 2). k is 1 in this example. Finally, the master machine selects k features using the Greedy algorithm (Step 3).

8–9 of Algorithm 2) are for getting a better approximation factor in theory. Figure 1 illustrates the vertical distribution of data in Algorithm 2 using the randomized composable core-set setting.

The next two theorems provide theoretical guarantees of the DDisMI method. Because of space limitations, we include full proofs in the supplemental material.

Theorem 1. *For any constant $0 < \epsilon \leq \frac{1}{4}$, Algorithm 2 is a $\frac{1-\epsilon}{4}$-approximation algorithm for diversity maximization problem with a high probability (e.g. probability $1-e^{-\Omega(\epsilon k)}$) using $m \geq 6/\epsilon$ machines.*

Proof sketch. *Let* OPT *be the optimum set of k points with the maximum diversity. Define OPT_ℓ to be $\text{OPT} \cap T_\ell$, the optimum solution points that are sent to machine ℓ. We prove that among the set of selected points in the first phase, $\cup_{\ell=1}^{m} S_\ell$, there is a benchmark set A of k points associated with and close to the k points of* OPT*, and consequently with a* $\text{Div}(A)$ *comparable to* $\text{Div}(\text{OPT})$. *Since we know that* Greedy *is a centralized $\frac{1}{2}$-approximation algorithm for diversity maximization, in total we will have a constant approximation guarantee for the* DDisMI *algorithm. We expect k/m optimum points in each machine, and with $m \geq 6/\epsilon$, using concentration bounds we can prove that with a high probability there are not more than $O(\epsilon k)$ items in each OPT_ℓ. We can use the optimum points that are selected by the machines ($\text{OPT} \cap (\cup_{\ell=1}^{m} S_\ell)$) as part of the high diversity benchmark set A. Suppose point $p \in \text{OPT}$ is sent to machine ℓ and is not selected. Instead, points $S_\ell = \{p_1, p_2, \cdots, p_k\}$ are selected in this machine. By definition of* Greedy*, we know that $\sum_{j=1}^{i-1} \text{Dist}(p, p_j) \leq \sum_{j=1}^{i-1} \text{Dist}(p_i, p_j)$. Summing all these inequalities for $2 \leq i \leq k$ implies that:*

$$\sum_{i=1}^{k-1} (k-i)\text{Dist}(p, p_i) \leq \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \text{Dist}(p_i, p_j) = \text{Div}(S_\ell) \tag{5}$$

On the left hand side, we have $\binom{k}{2}$ distances (by considering coefficients) from p to points in S_ℓ, and on the right hand side, there are all the $\binom{k}{2}$ pairwise distances in the set S_ℓ that is the diversity of set S_ℓ as well. Let τ be the maximum average distance of pairs of points in selected sets. In other words, τ is $\max_{1 \leq \ell \leq m} \frac{\text{Div}(S_\ell)}{\binom{k}{2}}$. One way to interpret the above inequality is that some weighted average of distances from p to points in S_ℓ is upper bounded by τ. By some algebraic computations, we imply that there should be at least $\Omega(\epsilon k)$ points in S_ℓ with distance at most $(1+\epsilon)\tau$ to p. We can use one of these points to represent p in A. So every optimum point is either in A or represented by a close point (distance at most $(1+\epsilon)\tau$) in it. By triangle inequality, we have $\text{Div}(A) \geq \text{Div}(\text{OPT}) - 2(1+\epsilon)\tau\binom{k}{2}$. *Since the final solution has a diversity at least half of* $\text{Div}(A)$ *and at the same time at least $\binom{k}{2}\tau$, we have* $\text{Div}(S) \geq \max\{\frac{\text{Div(OPT)}}{2} - \binom{k}{2}(1+\epsilon)\tau, \binom{k}{2}\tau\} \geq \frac{(1-\epsilon)\text{Div(OPT)}}{4}$. □

Algorithm 2: DDisMI

```
1  Input: $\mathbb{N}$, $k$, number of machines $m$.
2  Output: Set $S \subset \mathbb{N}$ with $|S| \leq k$.
3  $S \leftarrow \emptyset$;
4  Randomly partition $\mathbb{N}$ into $\{T_\ell\}_{\ell=1}^{m}$;
5  forall $1 \leq \ell \leq m$ do
6      $S_\ell \leftarrow$ output of Greedy($T_\ell$, $k$);
7  $S \leftarrow$ output of Greedy($\cup_{\ell=1}^{m} S_\ell$, $k$);
8  if $\exists \ell : \text{Div}(S_\ell) > \text{Div}(S)$ then
9      $S \leftarrow \arg\max_{S_\ell} \text{Div}(S_\ell)$;
10 Return $S$;
```

We are ready to slightly change the algorithm, and improve the approximation guarantee from $\frac{1}{4}$ to $\frac{8}{25}$. We note that although the analysis becomes more complex, the algorithm remains simple, and highly efficient. With a slight change in line 4 of Algorithm 2, instead of sending each point to a random machine, we pick C random machines (for a large enough C), and send the point to all of them. We call C the multiplicity factor since each item is sent to C multiple machines, and the rest of the algorithm remains the same. We prove that the approximation guarantee improves to $\frac{8}{25} - \epsilon$ where ϵ is sub-constant in terms of m, k, and C.

Theorem 2. *For every $\epsilon > 0$, Algorithm 2 with multiplicity $C = \frac{\ln(1/\epsilon)}{\epsilon}$ is a $(\frac{8}{25} - \epsilon)$-approximation algorithm in expectation (over the random partitioning of points) for diversity maximization problem.*

Empirical Study and Results

We have so far focused on formalizing the feature selection problem as a diversity maximization problem, and proving the theoretical foundations of our approach. We now turn to show that this method performs well in practice.

First, we compare the proposed method with the state-of-the-art centralized feature selection algorithms and demonstrate that, although it is a distributed method, it achieves consistently comparable results to the widely used centralized methods and even outperforms some of them. Next, we

Table 1: Comparison of different single-machine methods with GREEDY and DDISMI using the SVM classifier.

	MIQ	JMI	mRMR	SpecCMI	QPFS	CMI	Fisher Score	reliefF	GREEDY	DDISMI
Colon	78.1±5.3	83.1±2.5	79.8±2.3	72.2±4.1	78.0±2.8	69.7±5.4	81.3±3.2	71.7±5.1	**84.4±3.2**	83.1±3.2
Leukemia	92.7±4.8	88.7±3.9	**99.8±0.8**	89.5±3.8	82.0±4.0	82.0±4.0	99.7±0.6	98.9±1.1	96.0±2.0	96.1±2.3
Lung-discrete	82.1±4.2	90.9±2.3	90.7±1.7	84.1±8.7	91.4±2.6	83.8±6.7	88.7±5.5	80.8±6.6	**92.1±1.0**	91.5±1.7
Lung	86.8±2.5	90.4±2.9	**96.7±0.6**	95.2±0.8	96.6±0.7	96.3±0.8	89.8±4.2	92.2±1.9	96.4±1.0	95.9±1.4
Lymphoma	79.6±5.0	95.8±1.2	96.7±0.6	90.9±2.9	95.6±3.0	91.9±4.2	88.9±4.7	90.6±5.5	**97.8±1.1**	**97.8±1.1**
NCI9	30.2±4.5	74.0±3.5	74.4±2.9	51.2±5.4	25.0±5.4	52.2±7.3	64.8±4.6	50.7±5.0	**83.0±5.6**	82.2±6.3
Promoter	71.4±10.3	85.6±2.9	85.9±3.0	81.9±1.7	85.9±2.5	77.5±6.6	85.5±3.4	**86.7±3.7**	85.7±3.3	85.7±2.8
Srbct	82.7±8.5	80.8±5.5	99.6±0.6	98.6±1.9	**99.7±0.6**	99.3±0.8	98.7±1.6	94.2±5.1	99.2±1.2	98.7±1.0
TOX_171	61.4±5.2	57.4±7.9	83.8±5.8	77.4±7.0	72.5±5.7	88.4±7.5	75.1±5.2	79.5±18.3	88.3±3.5	**88.8±4.2**
Multi-features	85.5±9.1	83.8±10.5	96.0±2.2	95.6±3.0	96.1±0.8	95.5±2.5	95.6±2.2	**96.7±1.8**	96.5±0.7	96.3±0.8
Optdigits	81.1±22.3	96.7±2.0	96.3±2.2	96.3±2.3	95.9±2.8	85.4±15.1	96.5±2.0	95.4±4.7	**96.7±1.7**	96.6±1.8
USPS	92.6±3.8	90.9±3.7	90.0±2.6	86.9±2.6	92.5±4.1	93.3±3.2	88.5±8.0	89.7±5.1	**93.8±2.4**	**93.8±2.4**
PCMAC	70.3±0.9	88.0±1.7	88.5±1.3	87.9±1.4	87.8±1.3	89.2±1.7	88.5±2.3	71.5±1.3	89.8±2.0	**89.8±1.9**
RELATHE	67.1±2.2	79.8±3.1	82.5±2.8	79.9±3.1	79.2±3.0	**85.6±3.5**	76.2±6.6	63.0±3.5	85.0±4.3	84.7±4.6
Musk2	90.2±0.1	90.4±0.1	90.3±0.1	90.3±0.1	**90.6±0.2**	90.5±0.1	90.4±0.1	90.4±0.6	90.2±0.1	90.3±0.1
WarpAR10P	83.4±8.4	85.6±6.2	93.0±2.8	88.2±4.8	94.9±2.8	**94.9±2.2**	85.1±9.6	55.9±10.8	92.7±4.5	93.8±4.0
Pixraw10P	94.1±2.6	97.2±1.0	98.0±0.5	90.0±2.5	95.9±4.1	**98.5±0.7**	94.8±5.5	65.9±13.3	97.8±2.6	97.9±2.5
WarpPIE10P	96.9±4.4	97.3±1.5	96.4±0.9	95.0±1.1	96.9±1.1	98.4±1.3	96.2±1.9	94.9±3.5	98.7±1.0	**98.8±0.8**
Yale	52.4±6.5	66.3±5.9	72.2±3.0	63.0±7.3	**73.8±5.5**	72.6±4.5	70.8±3.3	55.4±9.2	71.3±4.7	70.8±5.3
W/T/L	19/0/0	14/4/1	11/4/4	17/1/1	12/2/5	11/1/7	14/3/2	15/1/3	2/11/6	–

present an empirical evaluation to show that the distributed version of our method (DDISMI) is tens of times faster than its centralized variant (GREEDY). Note that GREEDY itself is as fast as state-of-the-art centralized methods due to its efficient greedy strategy. Then we demonstrate the advantages of the greedy approach over the existing local search method for diversity maximization in terms of performance and running time. After that, we investigate the defined objective function by studying the effect of λ value on the results. Finally, to validate the quality of the objective function, we show a high correlation between the objective value and the classification accuracy on two small-sized datasets. Before elaborating on the empirical results, it should be mentioned that unlike the GREEDY algorithm which arbitrarily selects the first feature, in the implementation, we select the one that has the maximum MI with the class labels vector.

Comparison to the state-of-the-art feature selection methods. In this section, we compare the quality of various centralized feature selection methods with the proposed distributed (DDISMI) and centralized (GREEDY) methods. In order to test the sensitivity of our method to the structure of the dataset, we have used several datasets from a variety of domains with various number of features and instances in addition to the classic datasets in the literature of feature selection. We have described these datasets in detail in the supplemental material.

We considered a variety of MI-based filter methods, namely Mutual Information Quotient (MIQ) (Ding and Peng 2005), Minimum Redundancy Maximum Relevance (mRMR) (Ding and Peng 2005), Joint Mutual Information (JMI) (Yang and Moody 1999), Spectral Conditional Mutual Information (SpecCMI) (Nguyen et al. 2014), Quadratic Programming Feature Selection (QPFS) (Rodriguez-Lujan et al. 2010), and Conditional Mutual Information (CMI)(Cheng et al. 2008) as baselines as well as non MI-based methods, fisher score (Duda, Hart, and Stork 2001) and reliefF (Robnik-Šikonja and Kononenko 2003). Note that prior papers have performed extensive studies (Brown et al. 2012) comparing these methods and we have chosen methods that achieve the best results in various domains. To test the quality of the selected features of each method, we feed them into a classifier method M and compare their classification accuracy. All of the experiments are performed with both SVM and 3-NN as the classifiers (M). Note that all of the methods are filter-based and hence are independent from the selection of M. The LIBSVM package (Chang and Lin 2011) is the underlying implementation of the linear SVM with its regularization factor set to 1. We change $|S|$ from 10 to $min\{100, n\}$, where n is the number of the features in each dataset. Finally, we report the average cross validation (CV) classification accuracy of each method on all of the experiments with different values of $|S|$. A 10-fold CV is being used for the datasets with more than 100 instances and for the others, we employ the leave-one-out CV. In order to compute the probabilities used in MIs and VIs we have discretized the continuous variables using the Minimum Description Length (MDL) method (Irani and Fayyad 1993) with 5 bins. The regularization factor of our algorithm (λ) is set to be 0.8 in all of our experiments. We elaborate on choosing the λ in the experiment about effect of λ value. In order to run DDISMI, we simulate the distributed setting on a single machine and each (simulated) machine only has access to its own feature vectors. Each machine is responsible for processing $\sqrt{nk}$ features when we have n features and want to select k of them. Thus we employ $\sqrt{n/k}$ machines for the first stage. Then, we merge the results of all of the machines together and then process them again, i.e., we select k features from all of these $k\sqrt{n/k} = \sqrt{nk}$ features. For the sake of reproducibility, we have provided all the codes in the supplemental material. Table 1 compares the SVM classification accuracy of

Table 2: Comparison of different single-machine methods with GREEDY and DDISMI using the 3-NN classifier.

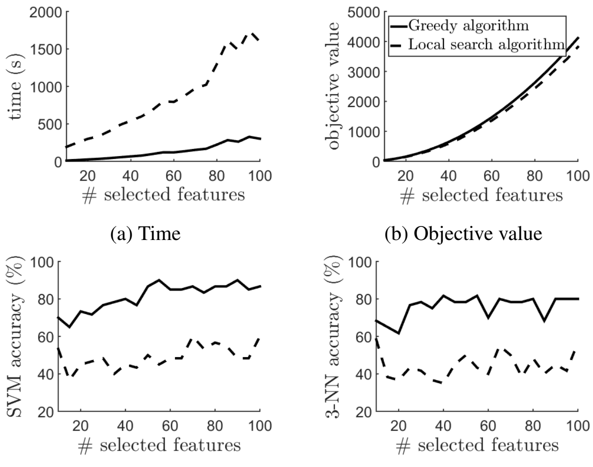
	MIQ	JMI	mRMR	SpecCMI	QPFS	CMI	Fisher Score	reliefF	GREEDY	DDISMI
Colon	70.6±5.9	86.8±1.3	85.9±1.6	66.8±6.8	82.4±2.0	70.5±5.8	86.0±1.7	83.2±1.5	**87.5±3.1**	87.0±2.5
Leukemia	88.3±2.7	88.5±1.5	**98.7±1.5**	81.0±2.9	82.4±2.1	82.4±2.1	98.4±1.3	97.4±1.1	91.7±1.6	91.6±1.8
Lung-discrete	83.5±4.0	**92.1±1.9**	91.3±1.5	83.9±5.1	90.8±1.5	84.8±2.4	88.6±3.9	82.3±4.9	91.0±1.9	90.7±1.7
Lung	84.0±3.0	88.4±4.0	**97.0±0.9**	94.8±1.1	96.9±0.9	95.7±0.8	88.7±3.3	92.6±2.2	95.9±1.2	95.9±1.6
Lymphoma	78.5±5.1	95.8±0.9	95.9±1.0	87.6±2.3	96.6±1.8	91.8±3.7	87.2±3.6	93.1±3.7	97.3±1.2	**97.7±1.4**
NCI9	34.9±2.9	76.1±3.0	75.9±2.9	53.4±4.4	37.3±4.0	59.5±5.9	75.9±5.7	60.1±4.3	**81.5±3.9**	80.1±5.0
Promoter	67.3±7.3	78.3±5.1	78.6±4.0	72.8±4.2	**78.6±3.9**	72.9±3.1	77.6±4.3	75.7±3.1	76.0±3.4	76.3±4.2
Srbct	77.9±7.5	70.6±3.0	**99.5±0.7**	97.9±2.4	99.5±0.8	99.0±1.0	98.8±1.4	92.0±5.3	97.3±1.8	98.2±1.6
TOX_171	61.0±4.9	47.3±8.5	76.0±3.4	70.6±2.7	68.7±3.3	72.2±3.8	66.5±3.7	66.4±6.2	**84.1±5.0**	83.4±5.3
Multi-features	48.9±1.8	62.2±18.7	90.8±3.3	91.0±2.9	87.8±4.7	91.8±2.0	89.8±3.4	**93.4±1.9**	89.6±5.2	89.1±6.8
Optdigits	80.5±24.9	**98.0±1.3**	97.8±1.6	97.7±1.8	97.3±2.5	85.8±16.1	97.8±1.6	96.7±4.0	97.8±1.4	97.7±1.5
USPS	94.0±3.7	91.8±3.1	89.6±2.7	86.6±6.4	93.3±4.6	94.3±3.7	88.8±7.8	91.2±5.0	95.3±2.7	**95.3±2.6**
PCMAC	55.3±1.1	79.5±1.0	82.7±1.6	81.8±1.1	82.0±1.1	82.7±1.4	83.9±1.9	64.7±4.3	**87.4±3.6**	86.9±3.6
RELATHE	59.8±2.4	72.5±4.5	74.5±4.0	70.7±5.0	68.6±5.6	75.5±4.4	69.8±5.4	61.9±3.3	79.5±5.5	**79.9±5.1**
Musk2	95.5±0.9	**96.2±0.5**	95.5±0.9	95.4±0.6	96.0±0.9	95.9±0.7	96.1±0.5	95.1±0.8	95.5±0.8	95.6±0.8
WarpAR10P	57.2±3.9	64.6±4.7	78.6±3.1	77.6±4.2	**87.0±2.7**	74.1±3.1	80.1±5.3	37.7±5.2	84.4±4.9	85.0±5.4
Pixraw10P	91.8±3.1	93.0±1.3	**98.1±0.6**	86.4±3.8	89.4±3.3	97.6±0.8	94.2±4.0	69.0±8.3	97.2±1.9	97.0±2.6
WarpPIE10P	86.2±10.5	89.9±4.3	95.1±1.3	93.9±1.6	95.6±1.1	**96.9±1.5**	95.4±2.2	90.6±6.8	96.3±1.5	96.4±1.4
Yale	47.9±4.9	56.5±4.8	**66.3±1.7**	54.7±2.4	62.7±4.7	57.3±2.2	65.9±2.5	55.3±4.4	62.3±5.0	61.3±4.9
W/T/L	18/1/0	14/1/4	10/1/8	16/2/1	12/1/6	13/1/5	12/3/4	16/1/2	3/12/4	–

the selected features. For each dataset (row), the results with higher accuracy and lower standard deviation are indicated with bold fonts. We examine the significance of the difference in the performance between two methods by two-tailed paired t-test at 5% significance level to decide if two feature selection methods have tied on a dataset. In the last row of Table 1, the performance of each method is compared with DDISMI in the number of datasets that DDISMI has won, tied or lost, respectively. From this table it can be inferred that, although DDISMI does not have access to global information like centralized methods, it outperforms them in many datasets and in some cases with slightly lower accuracy guarantees acceptable results. Also, Table 2 reports the results of the same experiment using the 3-NN classifier.

DDISMI speed-up in comparison with GREEDY. In this experiment, like before, we choose the number of the machines to be $\sqrt{n/k}$ in DDISMI. In order to test its speed-up in comparison with GREEDY, we employed some higher dimensional datasets from the order of tens of thousands to one million features. Since there is not any overhead of information-sharing between the machines, the speed-up is almost linear in terms of the number of the used machines. Our method selects 10 features from a dataset with 1.3 million feautres in 10 minutes using 368 machines. This is 164 times faster than running on a single machine which takes more than 27 hours. The results of this empirical study are summarized in Table 3.

Greedy vs. local search. In this experiment, we compare the greedy and the local search (Indyk et al. 2014) approaches for maximizing diversity in a distributed setting, to show that the greedy method not only achieves a higher objective value, but also performs better regarding classification accuracy and running time. The greedy method is much faster because it is linear in terms of the number of selected features and unlike the local search method, it does not need to converge. The results of this experiment for the NCI9 dataset are illustrated in Figure 2. Another dataset is also investigated in the supplemental material.

Effect of λ value. Another important issue that should be investigated is the relation of the two criteria in the defined objective function. We run the experiments of this part on the TOX_171 dataset to observe how changing the λ affects the classification results. Figure 3 (a) illustrates that the performance of each MI term ($\lambda = 0$) or VI term ($\lambda = 1$) in equation 4, individually, are not comparable with the performance of their linear combination. However, finding the optimal value of the λ is another challenge. Figure 3 (b) shows that regardless of the number of the selected features, we



(a) Time

(b) Objective value

(c) SVM accuracy

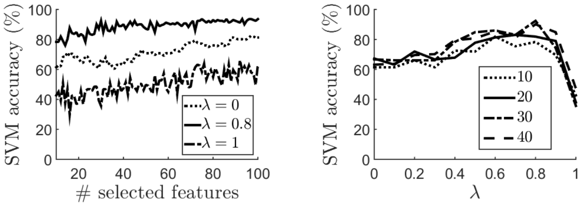
(d) 3-NN accuracy

Figure 2: Greedy vs. Local Search on the NCI9 dataset.

Table 3: Speed-up of the DDISMI method in comparison with GREEDY. “d”, “h” and “m” stand for day, hour and minute, respectively.

name	# features (n)	# instances (d)	# classes	# selected features	# machines	GREEDY time	DDISMI time	speed-up
Brain cell types (2015)	45771	1772	23	10	68	6.0m	0.2m	29.6
				20	48	12.4m	0.6m	22.3
				50	30	31.3m	2.3m	13.9
				100	21	1h 2.6m	6.2m	10.1
				200	15	2h 3.1m	20.9m	5.9
Dorothea (train part) (Guyon et al. 2004; Bache and Lichman 2013)	100,000	800	2	10	100	6.6m	0.1m	47.6
				20	71	12.6m	0.4m	31.6
				50	45	29.8m	1.4m	21.8
				100	32	1h 1.4m	4.2m	14.8
				200	22	2h 7.1m	11.3m	11.3
Binary news20 (Keerthi and DeCoste 2005; Chang and Lin 2011)	1,355,191	19996	2	10	368	1d 3h 36.4m	10.1m	164.1
				20	260	2d 7h 45.1m	28.0m	119.6
				50	165	6d 0h 20.9m	1h 52.4m	77.1
				100	116	12d 15h 36.2m	5h 29.9m	55.2
				200	82	23d 13h 48.6m	14h 13.6m	39.8

have a global maxima around the optimal λ, which suggests that it should be obtained from the intrinsic structure of the dataset. In this experiment, we have tested different λ values (from 0 to 1 with 0.1 steps) and it can be seen that 0.8 is a proper choice for λ. According to this result, we have set $\lambda = 0.8$ in all of the experiments but it should be mentioned that DDISMI performs much better when the λ parameter is adjusted for each dataset separately, e.g, it achieves 99.9 % accuracy when the λ is 0.4 on the Leukemia dataset.

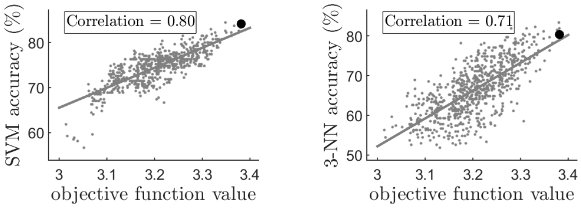


(a)

(b)

Figure 3: Effect of λ value on the classification accuracy in the TOX_171 dataset. (a) Accuracy of each criterion individually. (b) Finding optimal value of λ while selecting different number of features (10, 20, 30, 40).

Validating the objective function. One of the major questions regarding a new objective function is its relation with the classification accuracy. To test this issue, we select two small-sized datasets (small in the number of the features) so that we can evaluate all the possible combinations of the features. In this experiment, we compare the classification accuracy for all the 3-combinations and 4-combinations of features against their corresponding objective value with the λ parameter equal to 0.5. In Figure 4, each small grey dot represents the classification accuracy on a 4-combination of the features from the Heart (Bache and Lichman 2013) dataset, which has 13 features. The large black point is the solution of GREEDY and the line is the regression line. We observed that the objective value and the classification accuracy are highly correlated. Also, the solution of GREEDY has a high objective value as well as a high classification accuracy. The remaining results of this experiment can be found in the supplementaries.



(a) SVM accuracy

(b) 3-NN accuracy

Figure 4: Relation of classification accuracy and objective function value for 4-combinations of the Heart dataset.

We have done our experiments with two main goals in our mind. The first goal was to show that our distributed method can be compared with centralized methods. The second goal was to show why our method works. In order to satisfy the second goal, we have shown that both the objective function and the used greedy algorithm for maximizing it have a direct impact on the quality of the selected features. The impact of the greedy algorithm can directly be inferred from the results of the Greedy vs. local search experiment, however, the objective function’s effects has been shown implicitly via comparing it with the state-of-the-art methods, mainly because there are greedy-based algorithms such as mRMR in this comparison which were beaten by our method in various datasets due to their different objective function (but the same maximization procedure).

Conclusions

In this paper, we presented a new approach for vertically distributed feature selection based on diversity maximization and introduced a new objective function for this purpose based on a metric distance function. We showed that our distributed method is consistently comparable to the state-of-the-art centralized methods and it can handle ultrahigh-dimensional datasets very fast and efficient. Also, it handles the redundancy of the features very well, which is harder in a distributed setting and more crucial for ultrahigh-dimensional datasets. Moreover, we prove that our algorithm achieves a small constant factor approximation solution. For future work, achieving a distributed $\frac{1}{2}$-approximation algorithm is very appealing. For empirical studies, trying other metric objective functions to achieve better performance seems like a promising direction.

References

2015. Website: ©2015 allen institute for brain science. allen cell types database [internet]. `http://celltypes.brain-map.org`. Accessed: 2016-04-05.

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 *KDD*.

Aghamolaei, S.; Farhadi, M.; and Zarrabi-Zadeh, H. 2015. Diversity maximization via composable coresets. In *CCCG*, 38–48.

Bache, K., and Lichman, M. 2013. Uci machine learning repository.

Bhaskara, A.; Ghadiri, M.; Mirrokni, V.; and Svensson, O. 2016. Linear relaxations for finding diverse elements in metric spaces. In *NIPS*, 4098–4106.

Birnbaum, B., and Goldman, K. J. 2009. An improved analysis for a greedy remote-clique algorithm using factor-revealing lps. *Algorithmica* 55(1):42–59.

Bolón-Canedo, V.; Sánchez-Maroño, N.; and Alonso-Betanzos, A. 2015a. Distributed feature selection: An application to microarray data classification. *Appl. Soft Comput.* 30:136–150.

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.; Lee, H. C.; and Ye, Y. 2014. Max-sum diversification, monotone submodular functions and dynamic updates. *CoRR* abs/1203.6397v2.

Brown, G.; Pocock, A.; Zhao, M.-J.; and Luján, M. 2012. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research* 13(1):27–66.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.

Cheng, H.; Qin, Z.; Qian, W.; and Liu, W. 2008. Conditional mutual information based feature selection. In *KAM*, 103–107. IEEE.

Ding, C., and Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* 3(02):185–205.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2001. Pattern classification, a wiley-interscience publication john wiley & sons.

Goemans, M. 2015. Chernoff bounds, and some applications. `http://math.mit.edu/~goemans/18310S15/chernoff-notes.pdf`.

Greene, C. S.; Tan, J.; Ung, M.; Moore, J. H.; and Cheng, C. 2014. Big data bioinformatics. *Journal of cellular physiology* 229(12):1896–1900.

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

Guyon, I.; Gunn, S. R.; Ben-Hur, A.; and Dror, G. 2004. Result analysis of the NIPS 2003 feature selection challenge. In *NIPS*, 545–552.

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 *PODS*, 100–108. ACM.

Irani, K. B., and Fayyad, U. M. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, 1022–1029.

Keerthi, S. S., and DeCoste, D. 2005. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research* 6:341–361.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2016. Feature selection: A data perspective. *CoRR* abs/1601.07996.

Mirrokni, V. S., and Zadimoghaddam, M. 2015. Randomized composable core-sets for distributed submodular maximization. In *STOC*, 153–162.

Moran-Fernandez, L.; Bolón-Canedo, V.; and Alonso-Betanzos, A. 2015. A time efficient approach for distributed feature selection partitioning by features. In *CAEPIA*, 245–254.

Nguyen, X. V.; Chan, J.; Romano, S.; and Bailey, J. 2014. Effective global approaches for mutual information based feature selection. In *ACM SIGKDD*, 512–521. ACM.

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.

Peralta, D.; del Río, S.; Ramírez-Gallego, S.; Triguero, I.; Benitez, J. M.; and Herrera, F. 2015. Evolutionary feature selection for big data classification: A mapreduce approach. *Mathematical Problems in Engineering* 501:246139.

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

Rodriguez-Lujan, I.; Huerta, R.; Elkan, C.; and Cruz, C. S. 2010. Quadratic programming feature selection. *The Journal of Machine Learning Research* 11:1491–1516.

Sharma, A.; Imoto, S.; and Miyano, S. 2012. A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Trans. Comput. Biology Bioinform.* 9(3):754–764.

Turk-Browne, N. B. 2013. Functional interactions as big data in the human brain. *Science* 342(6158):580–584.

Yang, H. H., and Moody, J. E. 1999. Data visualization and feature selection: New algorithms for nongaussian data. In *NIPS*, 687–702.

Zhao, Z.; Cox, J.; Duling, D.; and Sarle, W. 2012. Massively parallel feature selection: An approach based on variance preservation. In *ECML-PKDD*, 237–252.

Appendix

Omitted proofs

Proof of Theorem 1 We start by associating each point $p \in \text{OPT}$ with some point in the set of selected points $\cup_{\ell=1}^{m} S_\ell$. Since there are m machines, and each point is sent to a random machine independently, we expect $\frac{k}{m}$ points in each set OPT_ℓ. The size of OPT_ℓ is a sum of k random variables $|\text{OPT}_\ell| = \sum_{p \in \text{OPT}} X_p$ where X_p is 1 if $p \in \text{OPT}_\ell$ and zero otherwise. Since there are independent binary random variables, we can apply concentration bounds like Upper Tail bound in Theorem 4 of (Goemans 2015) and limit $|\text{OPT}_\ell|$ with high probability. We imply that $Pr[|\text{OPT}_\ell| > (1+\delta)\mu] < e^{-\frac{\delta^2}{2+\delta}\mu}$ where μ is $\frac{k}{m}$, and δ is set to $m\epsilon/3 - 1$ so we have $(1+\delta)\mu = \epsilon k/3$. Since m is at least $6/\epsilon$, we know δ is at least 1, and consequently the exponent $\frac{\delta^2}{2+\delta}$ is at least $(1+\delta)/6$. We conclude that $Pr[|\text{OPT}_\ell| > \epsilon k/3] < e^{-\frac{1+\delta}{6}\mu} = e^{-k\epsilon/18}$ which converges to zero exponentially as k grows. By taking the union bound for all values of ℓ, we can say with probability at least $1 - e^{-\Omega(\epsilon k)}$ that there are at most L points in OPT_ℓ for every $1 \le \ell \le m$.

We construct set $A \subset \cup_{\ell=1}^{m} S_\ell$ of k points that represent points of OPT as follows. We add each point $p \in \text{OPT} \cap (\cup_{\ell=1}^{m} S_\ell)$ to set A (such point p represents itself in A). For every point $p \in \text{OPT}$ that is not selected ($p \notin \cup_{\ell=1}^{m} S_\ell$), we find a selected point close to it. Suppose p is sent to machine ℓ, and is not selected. Let $S_\ell = \{p_1, \ldots, p_k\}$ be the points that machine ℓ selected with the same order (p_1 is selected first, and p_k is selected last). According to our greedy selection, we have the following inequalities for any point $p \in (\text{OPT}_\ell \cap T_\ell) \setminus S_\ell$.

$$\begin{gathered}
\textsc{Dist}(p, p_1) \le \textsc{Dist}(p_2, p_1) \\
\textsc{Dist}(p, p_1) + \textsc{Dist}(p, p_2) \le \textsc{Dist}(p_3, p_1) + \textsc{Dist}(p_3, p_2) \\
\cdots \\
\sum_{i=1}^{k-1} \textsc{Dist}(p, p_i) \le \sum_{i=1}^{k-1} \textsc{Dist}(p_k, p_i)
\end{gathered} \tag{6}$$

Summing above inequalities implies that:

$$\begin{aligned}
\sum_{i=1}^{k-1} (k-i)\textsc{Dist}(p, p_i) &\le \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \textsc{Dist}(p_i, p_j) \\
&= \textsc{Div}(S_\ell)
\end{aligned} \tag{7}$$

On the left hand side, we have $\binom{k}{2}$ distances (by considering coefficients) from p to points in S_ℓ, and on the right hand side, there are all the $\binom{k}{2}$ pairwise distances in set S_ℓ that is the diversity of set S_ℓ as well. Let τ be the maximum average distance of pairs of points in selected sets. In other words, τ is $\max_{1 \le \ell \le m} \frac{\textsc{Div}(S_\ell)}{\binom{k}{2}}$. One way to interpret the above inequality is that some weighted average of distances from p to points in S_ℓ is upper bounded by τ. Following, we show that the distance of p to at least $\epsilon|S_\ell|/3$ points in S_ℓ is upper bounded by $(1+\epsilon)\tau$. Otherwise, there are $a > (1-\epsilon/3)k$ points in S_ℓ with distance more than $(1+\epsilon)\tau$ from p. In the left hand side of Equation 7, at least $\binom{a}{2}$ of the $\binom{k}{2}$ distances are greater $(1+\epsilon)\tau$. So we have the following lower bound on left side of Equation 7:

$$\begin{aligned}
&\binom{a}{2}(1+\epsilon)\tau \ge \frac{((1-\epsilon/3)k)((1-\epsilon/3)k-1)}{2} \times (1+\epsilon)\tau \\
&= \binom{k}{2}\tau \times \frac{(1-\epsilon/3)k}{k} \times \frac{(1-\epsilon/3)k-1}{k-1} \times (1+\epsilon) \\
&= \binom{k}{2}\tau \times (1-\epsilon/3) \times (1-\epsilon/3 - \frac{\epsilon/3}{k-1}) \times (1+\epsilon) \\
&\ge \binom{k}{2}\tau \times (1-\epsilon/3) \times (1-\epsilon/3 - \frac{\epsilon/3}{2}) \times (1+\epsilon) \\
&> \binom{k}{2}\tau \ge \textsc{Div}(S_\ell)
\end{aligned} \tag{8}$$

where the first inequality holds by the lower bound on a, and the second to the last inequality holds since $\epsilon \le 0.25$. The rest of the equations are simple algebraic manipulations. Combining the above lower bound on $\sum_{i=1}^{k-1}(k-i)\textsc{Dist}(p, p_i)$ with Equation 7 implies a contradiction. So there should be at least $\epsilon k/3$ points in S_ℓ with distance at most $(1+\epsilon)\tau$ from p. Since there are at most $\epsilon k/3$ points in OPT_ℓ with high probability, we can find one distinct representative point $p' \in S_\ell$ for each point $p \in \text{OPT}_\ell \setminus S_\ell$ to add to A.

We conclude that there exists a set A of k points among the selected points $\cup_{\ell=1}^{m} S_\ell$ that represent the k points of OPT, and the distance of each optimum point and its representative in A is upper bounded by $(1+\epsilon)\tau$. Using the triangle inequality, we know that $\textsc{Div}(A) = \sum_{p',q' \in A} \textsc{Dist}(p', q') \ge \sum_{p,q \in \text{OPT}} \textsc{Dist}(p, q) - \textsc{Dist}(p, p') - \textsc{Dist}(q, q') \ge \textsc{Div}(\text{OPT}) - 2\binom{k}{2}(1+\epsilon)\tau$ where p' and q' are the representatives of p and q.

We know that the $\textsc{Greedy}$ algorithm is a centralized $\frac{1}{2}$-approximation for diversity maximization (Birnbaum and Goldman 2009). In line 7 of Algorithm 2, we find a set S with diversity at least half of diversity of A, $\frac{1}{2}\textsc{Div}(A) \ge \frac{1}{2}(\textsc{Div}(\text{OPT}) - 2\binom{k}{2}(1+\epsilon)\tau)$. Since in lines 8 and 9, we take the maximum diversity of this selected set and all m selected sets $\{S_\ell\}_{\ell=1}^{m}$, the diversity of the final output set will be at least $\max\{\binom{k}{2}\tau, \frac{\textsc{Div}(\text{OPT})}{2} - (1+\epsilon)\binom{k}{2}\tau\}$ which is at least $\frac{\textsc{Div}(\text{OPT})}{4(1+\epsilon)} \ge \frac{1-\epsilon}{4}\textsc{Div}(\text{OPT})$. □

Proof of Theorem 2 The proof is similar to the proof of Theorem 1, and we just focus on the new ideas that help us improve the approximation guarantee. We still want to show that there exists a set of k points $A \subset \cup_{\ell=1}^{m} S_\ell$ with a high diversity. We focus on machine 1, and look how it filters the optimum points. We define OPT_1^S to be the set of points that if they are sent to machine 1 they will be selected. Formally, OPT_1^S is $\{x | x \in \text{OPT} \;\&\; x \in Greedy(T_\ell \cup \{x\})\}$ where

$Greedy(B)$ is the result of running Algorithm 1 on set B. We define OPT_1^{NS} to be $\mathrm{OPT} \setminus \mathrm{OPT}_1^{S}$. We note that the two sets OPT_1^{S} and OPT_1^{NS} form a partitioning of optimum set OPT, and one can similarly define OPT_ℓ^{S} and OPT_ℓ^{NS} for any ℓ. Let τ be the average distance of points in S_1, i.e. $\tau \overset{def}{=} \textsc{Div}(S_1)/\binom{k}{2}$. First of all, we show that the distance between any pair of points $p, q \in \mathrm{OPT}_1^{NS}$ is upper bounded by 2τ. Using Equation 7, we know that $\sum_{i=1}^{k-1}(k-i)\textsc{Dist}(p,p_i) \le \textsc{Div}(S_\ell)$ where p_i is the i^{th} point selected in S_1. We also have a similar inequality for q: $\sum_{i=1}^{k-1}(k-i)\textsc{Dist}(q,p_i) \le \textsc{Div}(S_\ell)$. Summing up these two lower bounds and using triangle inequality implies that:

$$\begin{aligned}\binom{k}{2}\textsc{Dist}(p,q) &\le \sum_{i=1}^{k-1}(k-i)\textsc{Dist}(p,p_i) + \textsc{Dist}(q,p_i)\\ &\le 2\textsc{Div}(S_\ell)\end{aligned} \tag{9}$$

By definition of τ, we conclude that $\textsc{Dist}(p,q)$ is at most 2τ. Intuitively, this means that the optimum points omitted by each machine are not too far from each other (compared to the solution set of the machine). The next step is to show that the points that are selected from the viewpoint of machine 1, OPT_1^{S}, will be selected with high probability. Based on Lemma 3.2 of (Mirrokni and Zadimoghaddam 2015), each point p in OPT is either in OPT_1^{S} with probability at most $\frac{\ln(C)}{C} \le \epsilon$ or it will be among the selected points with high probability $1 - \frac{1}{C} \ge 1 - \epsilon$. This holds mainly because at least one of the C machines that contain p will select it if it has a minimum probability (ϵ) of being selected in a random one (machine 1) of them. Let OPT^{S} be the set of selected optimum points $\mathrm{OPT} \cap (\cup_{\ell=1}^{m} S_\ell)$. Define set B to be the intersection $\mathrm{OPT}^{S} \cap \mathrm{OPT}_1^{S}$. Since the probability of each point being in $\mathrm{OPT}_1^{S} \setminus B$ is at most ϵ, the probability that two of optimum points belong to this set is upper bounded by 2ϵ. Therefore the expected diversity of set $\mathbb{E}[\textsc{Div}(\mathrm{OPT}_1^{NS} \cup B)]$ is at least $(1-2\epsilon)\textsc{Div}(\mathrm{OPT})$. We use set $\mathrm{OPT}_1^{NS} \cup B$ as a benchmark to show that there exists a set A of at most k points among the selected points with high diversity. Set B is among the selected points, so we put points of B into A as well. Let k' be $|\mathrm{OPT}_1^{NS}|$. We select k' random points from S_1 and put them in A. We show that these random points are close to the points in OPT_1^{NS} and consequently can represent OPT_1^{NS} well. Formally, we show that for any point $p \in \mathrm{OPT}_1^{NS}$, a random point in S_1 has distance at most $3\tau/2$ from p. Since p would not be chosen by machine 1, if it were sent to that machine, we have that:

$$X = \sum_{i=1}^{k}(k-i)\textsc{Dist}(p,p_i) \le \binom{k}{2}\tau = \textsc{Div}(S)$$

We denote the above sum by X to simplify the analysis. We can upper bound another similar summation denoted by Y with triangle inequality:

$$\begin{aligned} Y &= \sum_{i=1}^{k}(i-1)\textsc{Dist}(p,p_i)\\ &\le \sum_{i=2}^{k}\sum_{j=1}^{i-1}\textsc{Dist}(p,p_j) + \textsc{Dist}(p_j,p_i)\\ &= \sum_{j=1}^{k-1}(k-j)\textsc{Dist}(p,p_j) + \sum_{i=2}^{k}\sum_{j=1}^{i-1} + \textsc{Dist}(p_j,p_i)\\ &= X + \textsc{Div}(S_1)\end{aligned} \tag{10}$$

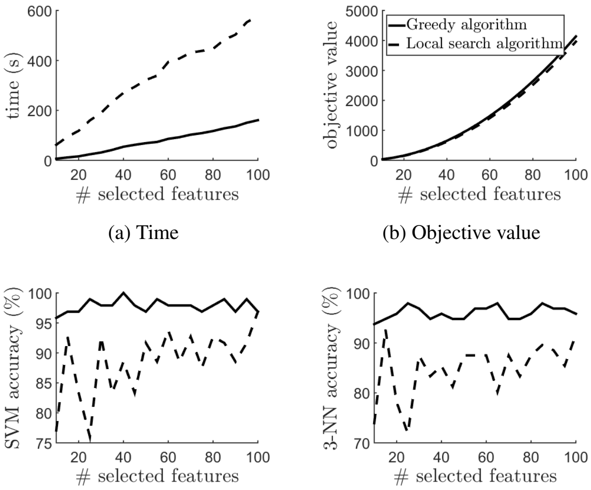
We also know that $X+Y$ is $k-1$ times the sum of distances from p to points in S_1, i.e. $X+Y = (k-1)\sum_{i=1}^{k}\textsc{Dist}(p,p_i)$. We conclude that $\sum_{i=1}^{k}\textsc{Dist}(p,p_i)$ is at most $(X+Y)/(k-1) \le (X+X+\textsc{Div}(S_1))/(k-1) \le 3\textsc{Div}(S_1)/(k-1)$. Therefore the expected distance of p to a random point in S_1 is upper bounded by $3\textsc{Div}(S_1)/(k(k-1)) \le 3\tau/2$.

We are ready to lower bound the expected diversity of A. For each pair of points p and q in OPT, the probability that both of them belong to set $\mathrm{OPT}_1^{NS} \cup B$ is at least $1-2\epsilon$. If both of them are in B, the same distance $\textsc{Dist}(p,q)$ will be counted in $\textsc{Div}(A)$ without any reduction. If both of them belong to OPT_1^{NS}, the distance $\textsc{Dist}(p,q)$ is at most 2τ, and they are replaced with two random points of S_1 in set A. Since the expected distance between two random points of S_1 is equal to τ (by definition of τ, we have a reduction (loss) of at most $2\tau - \tau = \tau$ in the distance. In the last case, we can assume that $p \in B$ and $q \in \mathrm{OPT}_1^{NS}$. Point p is present in A, and point q is replaced with a random point q' in S_1. We proved that the expected distance $\textsc{Dist}(q,q')$ is upper bounded by $3\tau/2$. Using triangle inequality we have $\textsc{Dist}(p,q') \ge \textsc{Dist}(p,q) - \textsc{Dist}(q,q') \ge \textsc{Dist}(p,q) - 3\tau/2$. We conclude that $\mathbb{E}[\textsc{Div}(A)]$ is at least $(1-2\epsilon)\textsc{Div}(\mathrm{OPT}) - \binom{k'}{2}\tau - 3k'(k-k')\tau/2)$ because there are $\binom{k'}{2}$ pairs of distances between points of OPT_1^{NS}, and we incur a distance reduction of at most τ for them. There are also $k'(k-k')$ pairs of distances between points of OPT_1^{S} and B with a loss of at most $3\tau/2$. The term $\binom{k'}{2}\tau - 3k'(k-k')\tau/2$ is maximized at $k' = (3k-1)/4$, and its maximum value is $((3k-1)/4)^2\tau$. Therefore, $\mathbb{E}[\textsc{Div}(A)]$ is at least $(1-2\epsilon)\textsc{Div}(\mathrm{OPT}) - ((3k-1)/4)^2\tau$. Since $\textsc{Greedy}$ in the second phase provides a set S with at least half of the value of A, and then we take the set with higher diversity among S and the m selected sets in lines 8 and 9, we conclude that the expected diversity of the final solution is at least $\max\{\frac{(1-2\epsilon)\textsc{Div}(\mathrm{OPT}) - ((3k-1)/4)^2\tau}{2}, \binom{k}{2}\tau\}$. To find the approximation guarantee that this lower bound implies, we denote $\binom{k}{2}\tau$ by α. Since $k \ge 2$, we have $((3k-1)/4)^2\tau \le \frac{9}{8}\alpha$. So the expected value of the diversity of the output set is at least $\max\{\frac{(1-2\epsilon)\textsc{Div}(\mathrm{OPT}) - 9\alpha/8}{2}, \alpha\}$ which is at least $\frac{8}{25}(1-2\epsilon)\textsc{Div}(\mathrm{OPT}) \ge (\frac{8}{25} - \epsilon)\textsc{Div}(\mathrm{OPT})$. □

Figures and tables of experiments

We have summarized the information of the datasets of our experiments in Table 4.

Greedy vs. Local search. Here, we compare the solution of these two methods empirically in the same setting on the Lymphoma dataset. The results of this experiment for the NCI9 dataset were illustrated in the main text. Figure 5 (b) shows that the greedy method outperforms the other competitor in getting higher values of the objective function with different number of selected features. As we expected from previous experiments, this higher value results in higher classification accuracy (Figure 5 (c & d)). In addition, it can be observed from Figure 5 (a) that the greedy method is much faster as it does not need to converge, and it is linear in terms of the the number of features to be selected.



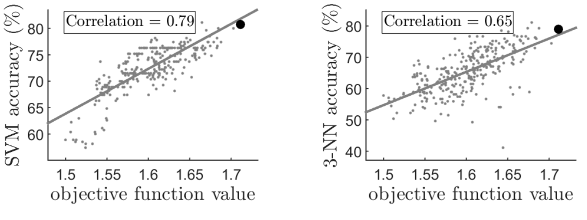
(a) Time

(b) Objective value

(c) SVM accuracy

(d) 3-NN accuracy

Figure 5: Greedy vs. Local Search on the Lymphoma dataset.

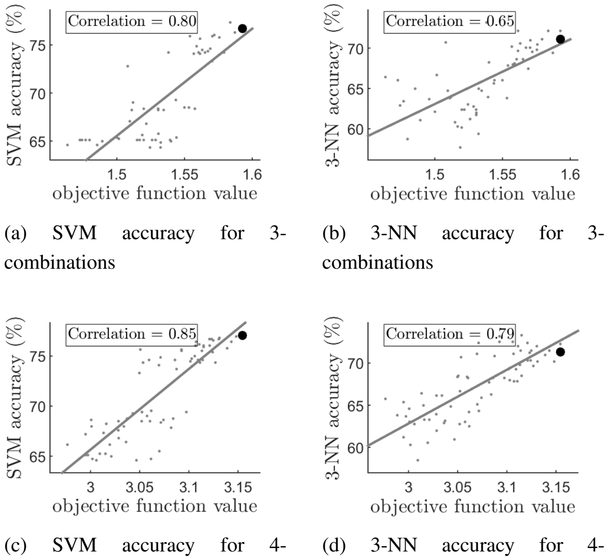


(a) SVM accuracy

(b) 3-NN accuracy

Figure 6: Relation of classification accuracy and objective function value for 3-combinations of the Heart dataset.

Validating the objective function. In this experiment, we compare the accuracy of SVM and 3-NN classifiers with 10-fold cross validation (CV) for all the 3-combinations and 4-combinations of features against their corresponding objective value on Diabetes (Chang and Lin 2011) and Heart (Bache and Lichman 2013) datasets. The Diabetes and Heart datasets have 8 and 13 features, respectively. In Figure 6, each small grey dot represents the classification accuracy on a 3-combination of the features from the Heart dataset. The large black point is the solution of GREEDY and the line is the regression line. Figure 7 illustrates the result of the same experiment for the Diabates dataset. It can be seen in Figures 6 and 7 that the objective value and the classification accuracy are highly correlated. Also, the solution of GREEDY which is shown by large black point has a high objective value as well as a high classification accuracy.



(a) SVM accuracy for 3-combinations

(b) 3-NN accuracy for 3-combinations

(c) SVM accuracy for 4-combinations

(d) 3-NN accuracy for 4-combinations

Figure 7: Relation of classification accuracy and objective function value for 3-combinations and 4-combinations of the Diabetes dataset.

Table 4: Summary of the datasets. All features accuracy indicates the accuracy of the SVM and 3-NN classifiers on all of the features.

name	# features (n)	# instances (d)	# classes	all features accuracy using SVM (%)	all features accuracy using 3-NN (%)	dataset type
Colon (Ding and Peng 2005)	2000	62	2	80.6	75.8	biological
Leukemia (Ding and Peng 2005)	7129	72	2	97.2	84.7	biological
Lung-discrete (Ding and Peng 2005)	325	73	7	87.7	87.7	biological
Lung (Li et al. 2016)	3312	203	5	87.7	96.6	biological
Lymphoma (Ding and Peng 2005)	4026	96	9	95.8	95.8	biological
NCI9 (Ding and Peng 2005)	9712	60	9	61.7	60.0	biological
Promoter (Bache and Lichman 2013)	57	106	2	84.0	71.7	biological
Srbct (Ding and Peng 2005)	2308	83	4	98.8	97.6	biological
TOX-171 (Li et al. 2016)	5748	171	4	97.1	72.5	biological
Multi-features (Bache and Lichman 2013)	649	2000	10	98.9	95.1	handwritten digits
Optdigits (Bache and Lichman 2013)	64	3823	10	96.9	98.6	handwritten digits
USPS (Chang and Lin 2011)	256	9298	10	94.8	97.2	handwritten text
PCMAC (Li et al. 2016)	3289	1943	2	89.5	77.6	text data
RELATHE (Li et al. 2016)	4322	1427	2	84.8	80.8	text data
Musk2 (Bache and Lichman 2013)	166	6598	2	96.7	96.7	chemical
WarpAR10P (Li et al. 2016)	2400	130	10	95.4	51.5	face image
Pixraw10P (Li et al. 2016)	10000	100	10	100.0	97.0	face image
WarpPIE10P (Li et al. 2016)	2420	210	10	99.5	96.2	face image
Yale (Li et al. 2016)	1024	165	15	76.4	66.1	face image