Overview

Multi-label feature selection problem is in need of scalable methods because of the rapid growth of the size of datasets.

Here we develop a theoretical modeling for this problem. We formulated it as a "submodular plus diversity" optimization problem and show that an approximation algorithm can be used to maximize this optimization problem in a distributed setting.

Main Contributions:
- Formulating the multi-label feature selection problem as a combinatorial optimization problem. Namely, as the maximization of the sum of a submodular subfunction and a sum-sum diversity function.
- Presenting a greedy algorithm for such a combinatorial optimization problem in the distributed and streaming settings and showing it achieves a constant factor approximation.
- Performing an empirical study on the multi-label feature selection method and comparing it to the state-of-the-art centralized feature selection methods.

Distributed Multi-label Feature Selection

<table>
<thead>
<tr>
<th>Samples</th>
<th>Features</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Properties of the dataset:
- A small number of samples
- A large number of features
Therefore we need a Filter method with Vertical Distribution of data.

A Theoretical Modeling

Goal: select non-redundant relevant features.

- Model the dis-similarity of features with a metric distance function.
- Model the relevance of features with a submodular function.

Set of features: $U = \{u_1, \ldots, u_n\}$
Set of labels: $L = \{l_1, \ldots, l_k\}$
The following is the dis-similarity/distance measure between pairs of features:

$$d(u_i, u_j) = 1 - \frac{|f(u_i, u_j)|}{M - |f(u_i, u_j)|}$$

The following is a submodular function that represents the relevance of a subset of features in the set of labels:

$$g(S) = \sum_{p \in P} \text{top}(M_f, S)$$

$s(S)$ is the joint entropy, $I_i$ is the mutual information, $M_f$ is the mutual normalized information, and top is the sum of the $p$ largest elements in the associated set.

The top function forms the criterion to select at least $p$ relevant features for each label. In the extreme cases of $p = 1$ and $p = \infty$, one or few features can dominate the formulation and prevent it from finding a good set of features.

Optimization Problem

Maximize the sum of a diversity function and a submodular function subject to a cardinality constraint.

$$\text{Maximize } \sum_{i=1}^{k} \sum_{j=1}^{n} d(u_i, u_j) + 1 - \frac{|L|}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} g(S)$$

$s(S)$ is the joint entropy, $l_i$ is a hyper-parameter.

Theoretical Problem

"Submodular plus Diversity"

Maximize the sum of a diversity function and a submodular function subject to a cardinality constraint.

Theoretical Contribution

Maximizing a "submodular plus diversity" function in distributed and streaming settings with a constant-factor approximation.

Algorithms

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. $S = \{\text{arg max}_{u_i} g(S \cup \{u_i\}) - g(S) + \sum_{l_j \in L} d(u_i, l_j)\}$
4. Return $S$.

Algorithm 2: 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. $S = \{\text{arg max}_{u_i} g(S \cup \{u_i\}) - g(S) + \sum_{l_j \in L} d(u_i, l_j)\}$
4. Return $S$.

Effect of $\lambda$

The legend values indicate the number of selected features.

Related Work

- Bredin et al (PDS12) show a half approximation for maximizing a "submodular plus diversity" function in the centralized setting.
- Abbasian-malick et al (AAAI17) show a quarter approximation for maximizing a diversity function in a distributed setting. They use this framework for single-label feature selection.
- Mavridis et al (STOC15) show a $0.27$-approximation for maximizing a submodular function in a distributed setting.
- Changpinyo et al (ICML13) consider the maximization of the sum of a submodular function and other diversity functions (sum-sum diversity, minimum spanning tree, and minimum distance).

Empirical Results

Speed-up

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Corel5k</th>
<th>RCV1</th>
<th>TMC2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>1000</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>500</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>250</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>125</td>
<td>12.5</td>
<td>12.5</td>
</tr>
</tbody>
</table>

Feature Selection Performance Compared to State-of-the-art

Centralized Methods

- Label power (LP) and binary relevance (BR) convert a multi-label dataset to one multiple single-label datasets.
- Relief (RF) and information gain (IG) are two methods for single-label feature selection.

Composable Core-sets

Distributed Setting

In the streaming setting, we have a machine that receives the data from a random stream.

Therefore, it can pick samples of data and use the greedy algorithm on them and store the selected features in the memory. Then when the stream is ended, it produces the final set of features by running the AlternatingGreedy algorithm on the stored features.

A composable core-set returns subsets which their union contains an approximate solution.

Here we use Randomized Composable Core-sets which mean the data is randomly shuffled.

Theoretical Result

We show that a randomized composable core-set finds a $\beta$-approximate solution in expectation.

Proof Idea:

Proof relies on the notion of $\beta$-niceness of an algorithm defined by Mavridis et al (STOC15). An algorithm is $\beta$-nice if the maximal gain of adding an element to the output is less than $\beta$ times the average contribution of the elements of the output.

This property shows that the output of an algorithm is "good enough" in the sense that adding other elements to its output does not increase the objective too much. This provides a theoretical bound for the optimum solution.

We show that the greedy algorithm is $\beta$-nice for this class of functions and using this, we conclude our result.