

# How to Use Submodular Dataset Creator

onigiri

June 21, 2014

## Contents

<b>1</b>	<b>Introductions</b>	<b>3</b>
<b>2</b>	<b>Common Usage of Left Side</b>	<b>3</b>
2.1	Path . . . . .	3
2.2	Number . . . . .	3
2.3	Type . . . . .	4
2.4	Execute and Save . . . . .	4
2.5	Save . . . . .	4
2.6	Text Box . . . . .	4
<b>3</b>	<b>Common Usage of Right Side</b>	<b>5</b>
3.1	Range Blank . . . . .	5
3.2	Value Blank . . . . .	5
3.3	Use Variable in a Blank . . . . .	5
<b>4</b>	<b>All</b>	<b>6</b>
<b>5</b>	<b>Undirected Cut</b>	<b>7</b>
<b>6</b>	<b>Directed Cut</b>	<b>8</b>
<b>7</b>	<b>Connected Detachment</b>	<b>9</b>
<b>8</b>	<b>Facility Location</b>	<b>11</b>
<b>9</b>	<b>Graphic Matroid</b>	<b>12</b>
<b>10</b>	<b>Binary Matrix Rank</b>	<b>14</b>
<b>11</b>	<b>Negative Symmetric Matrix Summation</b>	<b>15</b>
<b>12</b>	<b>Set Cover</b>	<b>16</b>

## 1 Introductions

This is the document for Submodular Dataset Creator. We recommend you to read 2, 4 and

## 2 Common Usage of Left Side

We will explain how to use functions on the left side of the application, which are common in any tabs.

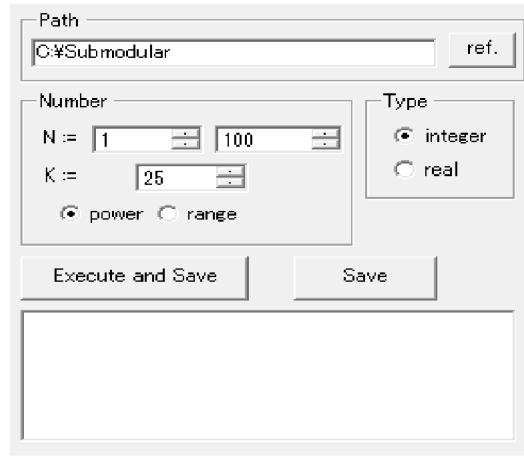
The screenshot shows a software window titled 'Submodular Dataset Creator'. It has a 'Path' section with a text box containing 'C:\\$Submodular' and a 'ref.' button. Below this is a 'Number' section with two rows: 'N :=' with a left spinner box at '1' and a right spinner box at '100', and 'K :=' with a spinner box at '25'. There are radio buttons for 'power' (selected) and 'range'. To the right is a 'Type' section with radio buttons for 'integer' (selected) and 'real'. At the bottom are 'Execute and Save' and 'Save' buttons, and a large empty rectangular area.

Figure 1: Common Functions

### 2.1 Path

We need to select a path in order to decide where we save data. You can directly type the path or select it by clicking "ref." button. If you type the path and it does not exist, new folder is automatically created.

### 2.2 Number

We can select  $N$  and  $K$ , where  $N$  is the cardinality of ground sets and  $K$  is the number of datasets for each cardinality. The application create datasets like Algorithm 1. Here,  $N_{\min}$  is the blank on left side of  $N$  and  $N_{\max}$  is the one on right side. If we choose "power",  $\text{Increment}(i) = 2 * i$ , and if we choose "range",  $\text{Increment}(i) = i + 1$ . For example, in the Figure 2, The application makes 25 ( $= K$ ) datasets whose cardinality of the ground sets are 1, 2, 4, 8, 16, 32 and 64 ( $= N$ ).

---

**Algorithm 1** "Create Datasets"

---

```
for ( $i = N_{\min}; i \leq N_{\max}; i = \text{Increment}(i)$ ) do  
  for ( $j = 0; j < K; j++$ ) do  
    Create a dataset whose the cardinality of the ground set is  $i$ .  
  end for  
end for
```

---

## 2.3 Type

To be Written!!!

## 2.4 Execute and Save

The application make datasets as stated in subsection 2.1, 2.2 and 2.3, and save something as stated in subsection 2.5. The files are created as `path/tabname/n_k`, where `path` is set in subsection 2.1, `tabname` is the name of tab which is currently selected (except All tab),  $n$  is the cardinality of the ground set and  $k$  is the index of the dataset.

## 2.5 Save

To be Written!!!

## 2.6 Text Box

In the bottom on the left side, there exist a text box. If the application has some trouble or it executes something, logs are output on this text box.

### 3 Common Usage of Right Side

We will explain how to use functions on the right side of the application, which are common in any tabs except the All tab.

#### 3.1 Range Blank

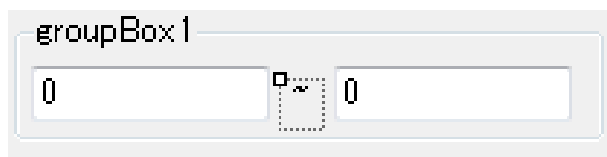


Figure 2: Range Blank

We first explain how to select a range of a random value. In order to select the range, you need to set two values, a minimum value and a maximum value. The left blank is for minimum value (min) and the right blank is for maximum value (max). Then, the range is defined by  $[\text{min} \dots \text{max}]$  (inclusive and inclusive) if the type is integer and  $[\text{min} \dots \text{max})$  (inclusive and exclusive) if the type is real (See 2.3 for the type).

From the definition above, if the type is real and  $\text{min} = \text{max}$ , then  $[\text{min} \dots \text{max})$  has no range. Therefore, it seems that  $\text{min} = \text{max}$  is not allowed. However, for convenience, the application always returns  $\text{min}(=\text{max})$  in this case.

#### 3.2 Value Blank

We next explain value blank. The range blank has two blanks but value blank has only one blank. If you type a value in the blank, then this value is set as the parameter.

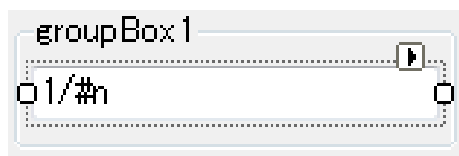


Figure 3: Value Blank

#### 3.3 Use Variable in a Blank

You can use variable  $n$ , where  $n$  is the cardinality of the ground set. For example, if you want to set the value as  $1/n$ , please type  $1/\#n$  or  $1/\#N$ . If the value should be integer but the calculated value is not integer, this value is rounded down.

## 4 All

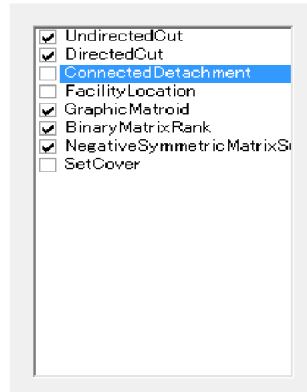


Figure 4: Candidates of datasets

In All tab, the application creates datasets which are checked at a time. There for if you want to make data with same condition, it is very useful. For the parameters on the left side of the application, like  $N$ ,  $K$ , path, the application uses the ones of the All's tab, but for the other parameters it uses the ones of each dataset's tab.

## 5 Undirected Cut

Let  $G = (V, E, w)$  be a complete undirected nonnegative weighted graph,  $\delta_w : V \rightarrow \mathbb{R}_{\geq 0}$  be an undirected weighted cut function,  $m$  be a modular function. Then,  $\delta_w + m$  is a submodular function. In the Undirected Cut tab, the application creates datasets for making such submodular functions.

We explain meanings of blanks in this tab.

- modular: This is the value of  $m$ .
- edge weight: This is the value of  $w$ . This value must be non-negative. In order to express  $w$ , the application makes a symmetric matrix. The  $(i, j)$ th entries of this matrix means that the weight of the edge from  $i$  to  $j$ . The application sets  $(i, i)$ th entries are 0.

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 1$  lines as follows.

```

n
m_1
⋮
m_n
w_11 ⋯ w_1n
⋮
w_n1 ⋯ w_nn
```

Here,  $w_{i1} \cdots w_{in}$  is separated by an empty character " ".

## 6 Directed Cut

Let  $G = (V, E, w)$  be a complete directed nonnegative weighted graph,  $\delta_w : V \rightarrow \mathbb{R}_{\geq 0}$  be a directed weighted cut function,  $m$  be a modular function. Then,  $\delta_w + m$  is a submodular function. In the Directed Cut tab, the application creates datasets for making such submodular functions.

- modular: This is the value of  $m$ .
- edge weight: This is the value of  $w$ . This value must be non-negative. In order to express  $w$ , the application makes a matrix. The  $(i, j)$ th entries of this matrix means that the weight of the edge from  $i$  to  $j$ . The application sets  $(i, i)$ th entries are 0.

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 1$  lines as follows.

```
n
m1
⋮
mn
w11 ⋯ w1n
⋮
wn1 ⋯ wnn
```

Here,  $w_{i1} \cdots w_{in}$  is separated by an empty character ” ”.



## 7 Connected Detachment

See [5] for details. Let  $G = (V, E)$  be a connected undirected graph,  $m$  be a modular function. For all  $X \subseteq V$ , let  $c(X)$  be the number of connected component of induced subgraph  $G(X)$  and  $e(X)$  be the number of edges which is incident with  $X$ . Then,  $m(X) - e(X) + c(V \setminus X) - 1$  is a submodular function. In the Connected Detachment tab, the application creates datasets for making such submodular functions.

- modular: This is the value of  $m$ .
- probability of edge: This is the probability  $p$  that an edges exists. The application makes random graphs with this  $p$ . For each  $(i, j)$  with  $i + 1 < j$ , the application connects  $i$  and  $j$  if and only if  $\text{RandDobule}([0..1)) < p$ . Note that since the graph is connected, it connects  $i$  and  $i + 1$  in advance.

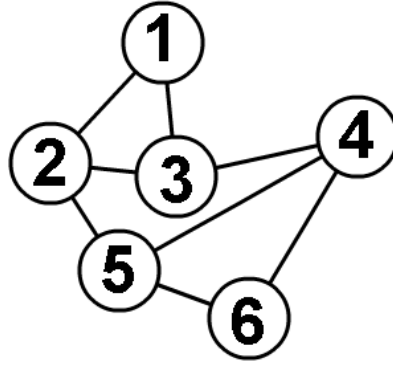


Figure 5: example graph

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 1$  lines as follows.

```

n
m1
⋮
mn
l1e11 ⋯ e1l1
⋮
lnen1 ⋯ enln

```

Here,  $l_i$  is the number of edges with incident with  $i$ , and each element is separated by empty character " ". Therefore, the graph is expressed by an adjacency list.

For example, if we are given a graph like 7, then the graph part of the output is as follows.

```
2 3
1 3 5
1 2 4
3 5 6
2 4 6
4 5
```

## 8 Facility Location

Let  $M \in \mathbb{R}^{n \times n}$  be a nonnegative matrix,  $m$  be a modular function. Let  $V := \{1, \dots, n\}$ . For any  $X \subseteq V$ , define

$$f(X) := m(X) + \sum_{i \in V} \max\{M_{ij} \mid j \in X\}$$

Then,  $f$  is a submodular function. In the Facility Location tab, the application creates datasets for making such submodular functions.

- modular: This is the value of  $m$ .
- matrix element: This is the value of  $M$ .

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 1$  lines as follows.

```

n
m1
⋮
mn
M11 ⋯ M1n
⋮
Mn1 ⋯ Mnn

```

## 9 Graphic Matroid

Let  $G = (V, E)$  be a graph. For all  $X \subseteq E$ , define  $r(X) := \max\{|Y| \mid Y \subseteq X, \text{ the induced subgraph } G(Y) \text{ does not have a cycle.}\}$ . This  $r$  is called the rank function of the graphic matroid which arises from  $G$ , and therefore  $r$  is a submodular function. For example, if  $G$  is a graph showed in Figure 9 then  $r(\{\{1, 3\}, \{2, 3\}\}) = 2$  and  $r(\{\{2, 5\}, \{4, 5\}, \{4, 6\}, \{5, 6\}\}) = 3$ . Let  $m$  be a modular function. For all  $X \subseteq V$ , define  $f(X) := r(X) + r(V \setminus X)$ . Then,  $f$  is also a submodular function. In the Graphic Matroid tab, the application creates datasets for making such submodular functions.

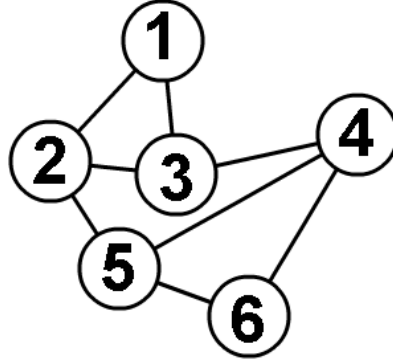


Figure 6: example graph

---

**Algorithm 2** "Create  $n$  edges"

---

```

for ( $i = 1; i \leq n; i = i + 1$ ) do
     $u_i := \text{Rand}(|V|)$ 
     $v_i := \text{Rand}(|V|)$ 
    connect  $u$  and  $v$ 
end for

```

---

- modular: This is the value of  $m$ .
- number of vertices: This is  $|V|$ . Let this value is  $n$ . Then the application uses Algorithm 2 to make  $n$  edges.

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 2$  lines as follows.

$$\begin{array}{l}
n \\
m_1 \\
\vdots \\
m_n \\
|V| \\
u_1 \ v_1 \\
\vdots \\
u_n \ v_n
\end{array}$$

Here,  $|V|$  is the number of vertices and  $u_i$  and  $v_i$  is the ones described in Algorithm 2.

## 10 Binary Matrix Rank

Let  $M^0, M^1 \in \mathbb{R}^{k \times n}$  be a binary matrix, i.e., a matrix whose elements are 0 or 1, and  $V := \{1, 2, \dots, n\}$ . Denote the  $i$ th column vector of the matrix  $M$  by  $M_i$ . For all  $X \subseteq V$ , define  $r(X) := \max\{|Y| \mid Y \subseteq X, (M_i)_{i \in Y} \text{ is linearly independent.}\}$ . Then,  $r$  is the rank function of  $M$  according to column, and it is a submodular function. Let  $m$  be a modular function. For all  $X \subseteq V$ , define  $f(X) := m(X) + r(X) + r(V \setminus X)$ . Then,  $f$  is also a submodular function. In the Binary Matrix Rank tab, the application creates datasets for making such submodular functions.

- modular: This is the value of  $m$ .
- probability of element: This is the probability of the  $(i, j)$ th element is 1 (and the others are 0).
- column length: This is  $k$  above.

Let  $n$  be the cardinality of the ground set. Then, output files have  $n + k + 2$  lines as follows.

```

n
m_1
⋮
m_n
k
M_11 ⋯ M_1n
⋮
M_k1 ⋯ M_kn
```

## 11 Negative Symmetric Matrix Summation

Let  $M \in \mathbb{R}_{\leq 0}^{n \times n}$  be a symmetric matrix whose elements are non-positive,  $V := \{1, 2, \dots, n\}$ ,  $m$  be a modular function. For all  $X \subseteq V$ , define  $f(X) := m(X) + \sum_{i,j \in X} M_{ij}$ . Then,  $f$  is a submodular function. In the Negative Symmetric Matrix Summation tab, the application creates datasets for making such submodular functions.

- modular: This is the value of  $m$ .
- matrix element: This is the value of  $M$ . This value must be non-positive.

Let  $n$  be the cardinality of the ground set. Then, output files have  $2n + 1$  lines as follows.

```

n
empty value
m1
⋮
mn
M11 ⋯ M1n
⋮
Mn1 ⋯ Mnn

```

## 12 Set Cover

Let  $G = (V, U, E)$  be a bipartite graph with  $|V|$ ,  $m_V$  be a modular function on  $V$ ,  $m_U$  be a non-negative modular function on  $U$  and  $C$  be a constant. For all  $X \subseteq V$ , define  $f(X) := m_V(X) + \varphi(N(X))$  is a submodular function. Here,  $N$  is a neighbor function, i.e., for all  $X \subseteq V$ ,  $N(X) := \{u \in U \mid \exists v \in X \{u, v\} \in E\}$  and  $\varphi$  is a concave function. In the Set Cover tab, the application creates datasets for making such submodular functions.

- modular: This is the value of the modular function on  $V$ .
- the cardinality of vertices: This is  $|V|$ .
- weight of element: This is the value of the modular function on  $U$ . This value is must be non-negative.
- matrix element: This is the probability  $p$  that an edge exists. For each  $(i, j)$  where  $i \in V, j \in U$ , the application connects  $i$  and  $j$  if and only if  $\text{RandDobule}([0..1)) < p$ . See Algorithm 3 for details.

---

### Algorithm 3 "Create random edges"

---

**Require:**  $p$  is the probability

```

for ( $i = 1; i \leq |V|; i = i + 1$ ) do
  for ( $j = 1; j \leq |U|; j = j + 1$ ) do
    if  $\text{RandomReal}() < p$  then
      connect  $i \in V$  and  $j \in U$ 
    end if
  end for
end for

```

---

Let  $n$  be the cardinality of the ground set. Then, output files have  $3n + 2$  lines as follows.

```

 $n$ 
 $m_{V_1}$ 
 $\vdots$ 
 $m_{V_n}$ 
 $|U|$ 
 $m_{U_1}$ 
 $\vdots$ 
 $m_{U_{|U|}}$ 
 $e_{11} \cdots e_{1l_1}$ 
 $\vdots$ 
 $e_{n1} \cdots e_{nl_n}$ 

```



Here,  $l_i$  is the number of edges with incident with  $i$ , and each element is separated by empty character " ". Therefore, the graph is expressed by an adjacency list.

## References

- [1] K. P. Bennett and E. J. Brendenstein. Duality and geometry in SVM classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 57–64. Morgan Kaufmann, 2000.
- [2] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. Technical report, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [3] D. J. Crisp and C. J. C. Burges. A geometric interpretation of  $\nu$ -SVM classifiers. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2000.
- [4] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [5] C. St. J. A. Nash-Williams. Connected detachments of graphs and generalized euler trails. *J. London Math. Soc.*, year =.
- [6] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.
- [7] A. Sideris and S. E. Castella. A proximity algorithm for support vector machine classification. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.
- [8] P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11:128–149, 1976.