Tutorial

Memetic Feature Selection Algorithm for Multi-label Classification

Published from *Information Sciences* 293(1), 1 February 2015

Information Sciences 293 (2015) 80–96

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Memetic feature selection algorithm for multi-label classification

Jaesung Lee, Dae-Won Kim*

School of Computer Science and Engineering, Chung-Ang University, 221, Heukseok-Dong, Dongjak-Gu, Seoul 156-756, Republic of Korea

ARTICLE INFO

Article history:
Received 17 March 2014
Received in revised form 15 August 2014
Accepted 12 September 2014
Available online 20 September 2014

Keywords:
Multi-label feature selection
Memetic algorithm
Local refinement

ABSTRACT

The use of multi-label classification, i.e., assigning unseen patterns to multiple categories, has emerged in modern applications. A genetic-algorithm based multi-label feature selection method has been considered useful because it successfully improves the accuracy of multi-label classification. However, genetic algorithms are limited to identify fine-tuned feature subsets that are close to the global optimum, which results in a long runtime. In this paper, we present a memetic feature selection algorithm for multi-label classification that prevents premature convergence and improves the efficiency. The proposed method employs memetic procedures to refine the feature subsets found through a genetic search, resulting in an improvement in multi-label classification. Empirical studies using various tests show that the proposed method outperforms conventional multi-label feature selection methods.

1. Introduction

Multi-label classification is a challenging problem that has emerged in several modern areas of application, such as text categorization [14], gene function classification [11], and the semantic annotation of images [1]. Let $W \subset \mathbb{R}^d$ denote an input space constructed from a set of features F, where $|F| = d$ and patterns drawn from W are assigned to a certain label subset $\lambda \subseteq Y$, where $Y = \{y_1, \ldots, y_\psi\}$ is a finite set of labels with $|Y| = \psi$. Thus, multi-label classification is the task of assigning unseen patterns to multiple labels. However, this is a difficult task because its efficacy can be varied according to the number of labels, features, patterns, and evaluation measures used to assess the quality of the predicted labels from different aspects [4,5,18,23,28,29,36,37].

Based on exhaustive experiments, researchers have reported that feature selection can improve the performance of multi-label classification [2,12,13,15,16,20,26,33–35]. Researchers have considered various approaches to performing feature selection for multi-label learning. Among them, the genetic-algorithm (GA) based multi-label feature selection method has shown strength in terms of classification performance [34] because it evaluates the fitness of the feature subsets using a multi-label classifier directly. However, owing to its inherent characteristics, a GA consumes enormous time to find a feature subset and sometimes may not find the optimum subset with sufficient precision; thus, it often converges to prematured solutions.

Recent studies on memetic algorithms (MAs) have demonstrated that they converge to high-quality solutions more efficiently than GAs for complex problems. Zhu et al. and Oh et al. presented a memetic algorithm-based feature selection that

* Corresponding author. Tel.: +82 2 820 5304; fax: +82 2 820 5301.
E-mail address: dwkim@cau.ac.kr (D.-W. Kim).

http://dx.doi.org/10.1016/j.ins.2014.09.020

Jaesung Lee and Dae-Won Kim

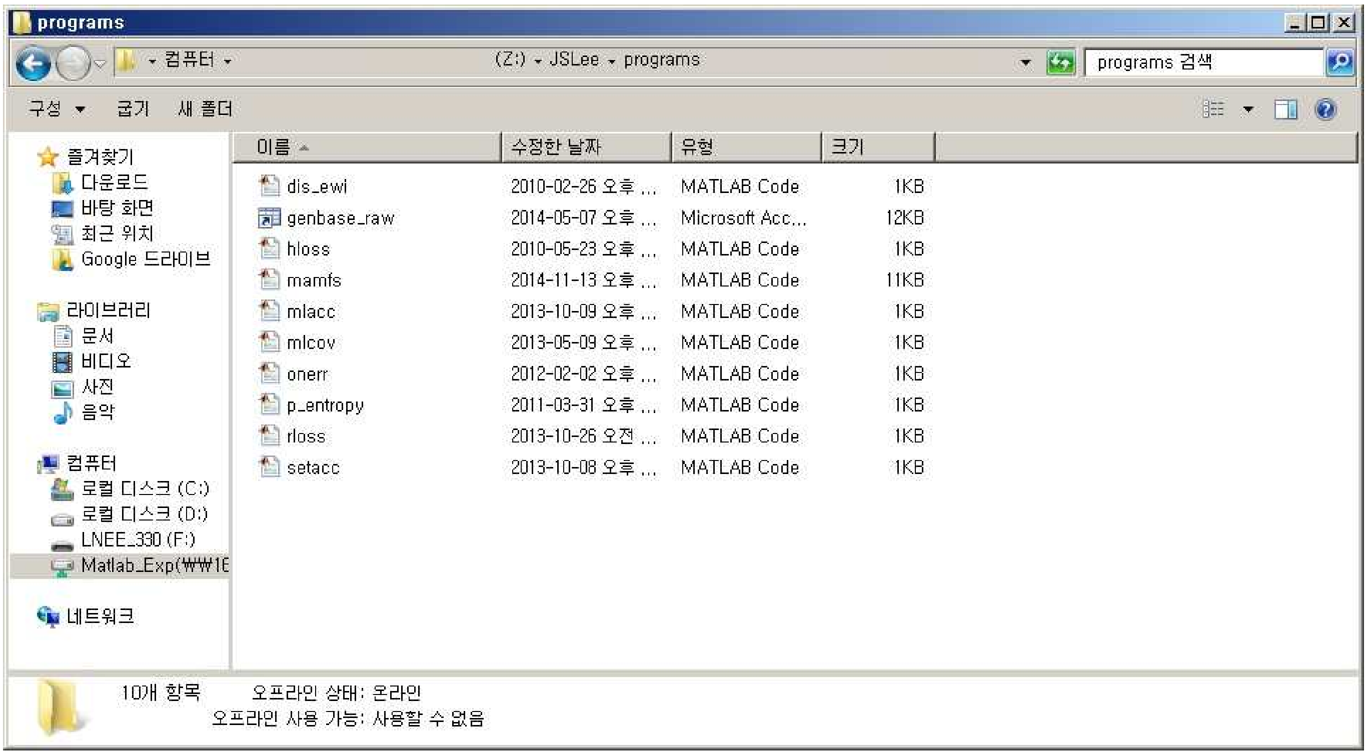
Department of Computer Science and Engineering, Chung-Ang University, Korea

9 November 2015

Abstract

The use of multi-label classification, i.e., assigning unseen patterns to multiple categories, has emerged in modern applications. A genetic-algorithm based multi-label feature selection method has been considered useful because it successfully improves the accuracy of multi-label classification. However, genetic algorithms are limited to identify fine-tuned feature subsets that are close to the global optimum, which results in a long runtime. In this paper, we present a memetic feature selection algorithm for multi-label classification that prevents premature convergence and improves the efficiency. The proposed method employs memetic procedures to refine the feature subsets found through a genetic search, resulting in an improvement in multi-label classification. Empirical studies using various tests show that the proposed method outperforms conventional multi-label feature selection methods.

Step 1. Download and unzip the file (programs.zip).



이름	수정한 날짜	유형	크기
dis_ewi	2010-02-26 오후 ...	MATLAB Code	1KB
genbase_raw	2014-05-07 오후 ...	Microsoft Acc...	12KB
hloss	2010-05-23 오후 ...	MATLAB Code	1KB
mamfs	2014-11-13 오후 ...	MATLAB Code	11KB
mlacc	2013-10-09 오후 ...	MATLAB Code	1KB
mlcov	2013-05-09 오후 ...	MATLAB Code	1KB
onerr	2012-02-02 오후 ...	MATLAB Code	1KB
p_entropy	2011-03-31 오후 ...	MATLAB Code	1KB
rloss	2013-10-26 오전 ...	MATLAB Code	1KB
setacc	2013-10-08 오후 ...	MATLAB Code	1KB

10개 항목 오프라인 상태: 온라인
오프라인 사용 가능: 사용할 수 없음

Note

A. In this example, I unzipped the downloaded files to "Z:\JSLee\programs" folder.

Check points

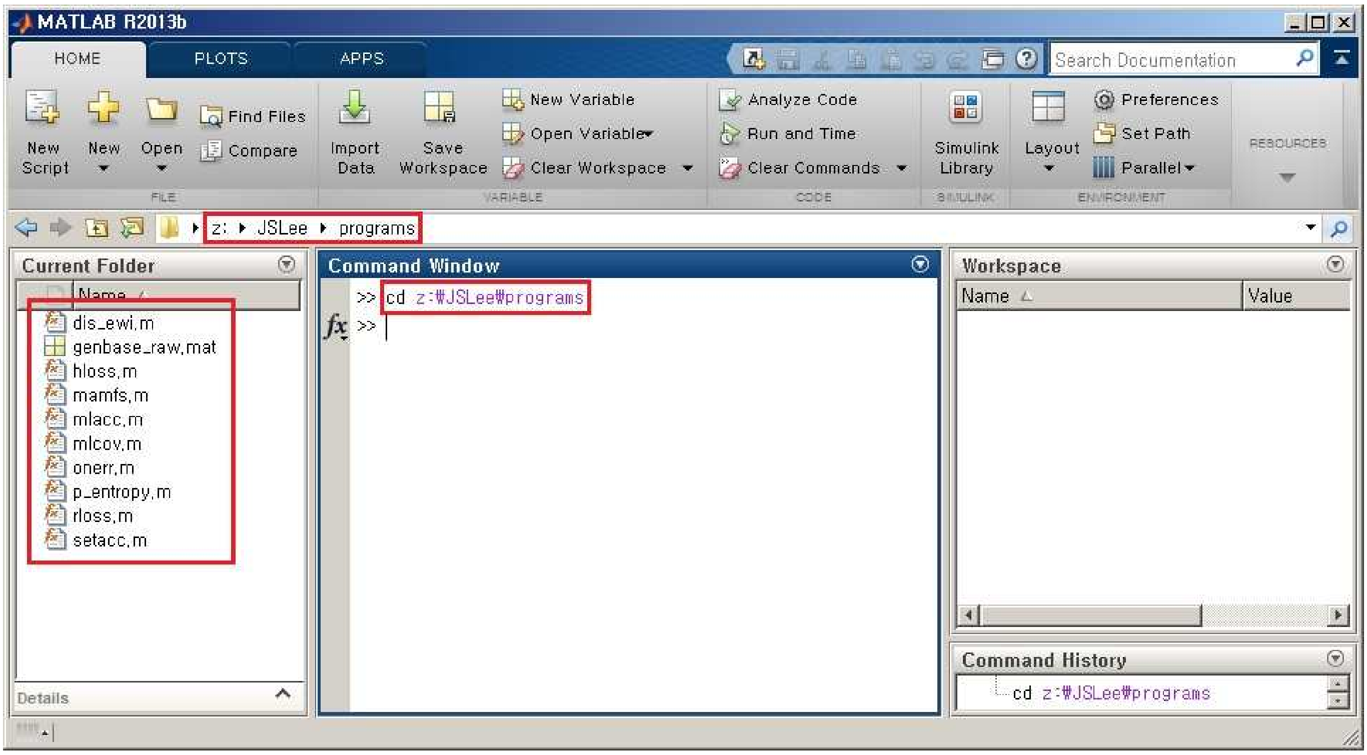
A. "programs.zip" file contains 10 files in total.

B. "genbase_raw.mat" file contains an example data set that is publicly-available on the web site "http://mulan.sourceforge.net/download.html" with below references:

* Mulan: Tsoumakas, G., Katakis, I., Vlahavas, I. (2010) "Mining Multi-label Data", Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.

* Genbase data set: S. Diplaris, G. Tsoumakas, P. Mitkas and I. Vlahavas. "Protein Classification with Multiple Algorithms," Proc. 10th Panhellenic Conference on Informatics (PCI 2005), pp. 448-456, Volos, Greece, November 2005.

Step 2. Execute MATLAB and move to the work folder.



```
>> cd z:₩JSLee₩programs
>> 
```

- dis_ewi.m
- genbase_raw.mat
- hloss.m
- mamfs.m
- mlacc.m
- mlcov.m
- onerr.m
- p_entropy.m
- rloss.m
- setacc.m

```
cd z:₩JSLee₩programs
```

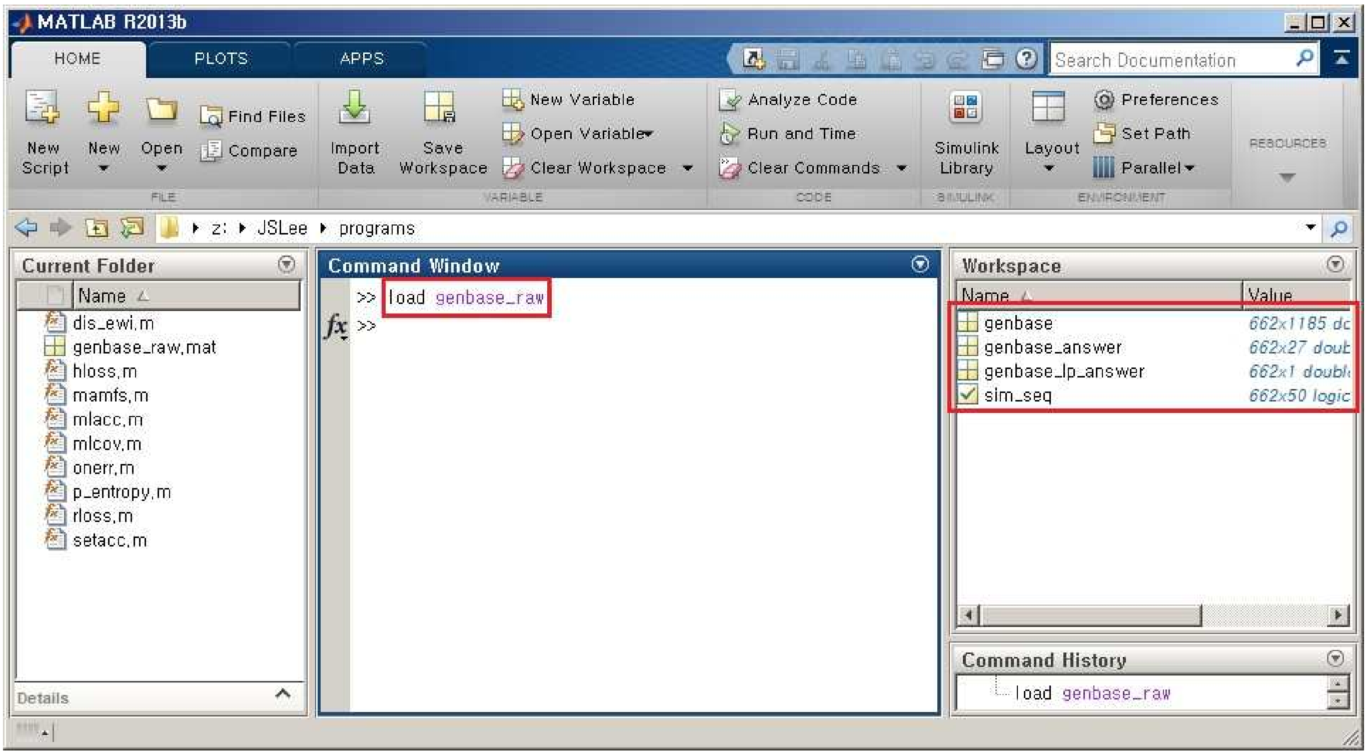
Note

A. To move the current folder of MATLAB, type "cd Z:\JSlee\programs" to "Command Window".

Check points

A. If you change the current folder correctly, you will see the unzipped files from "Current Folder" window.

Step 3. Load the example file (genbase_raw.m).



Current Folder

Name
dis_ewi.m
genbase_raw.mat
hloss.m
mamfs.m
mlacc.m
mlcov.m
onerr.m
p_entropy.m
rloss.m
setacc.m

Command Window

```
>> load genbase_raw
>>
```

Workspace

Name	Value
genbase	662x1185 dc
genbase_answer	662x27 douk
genbase_lp_answer	662x1 doubl
sim_seq	662x50 logic

Command History

```
load genbase_raw
```

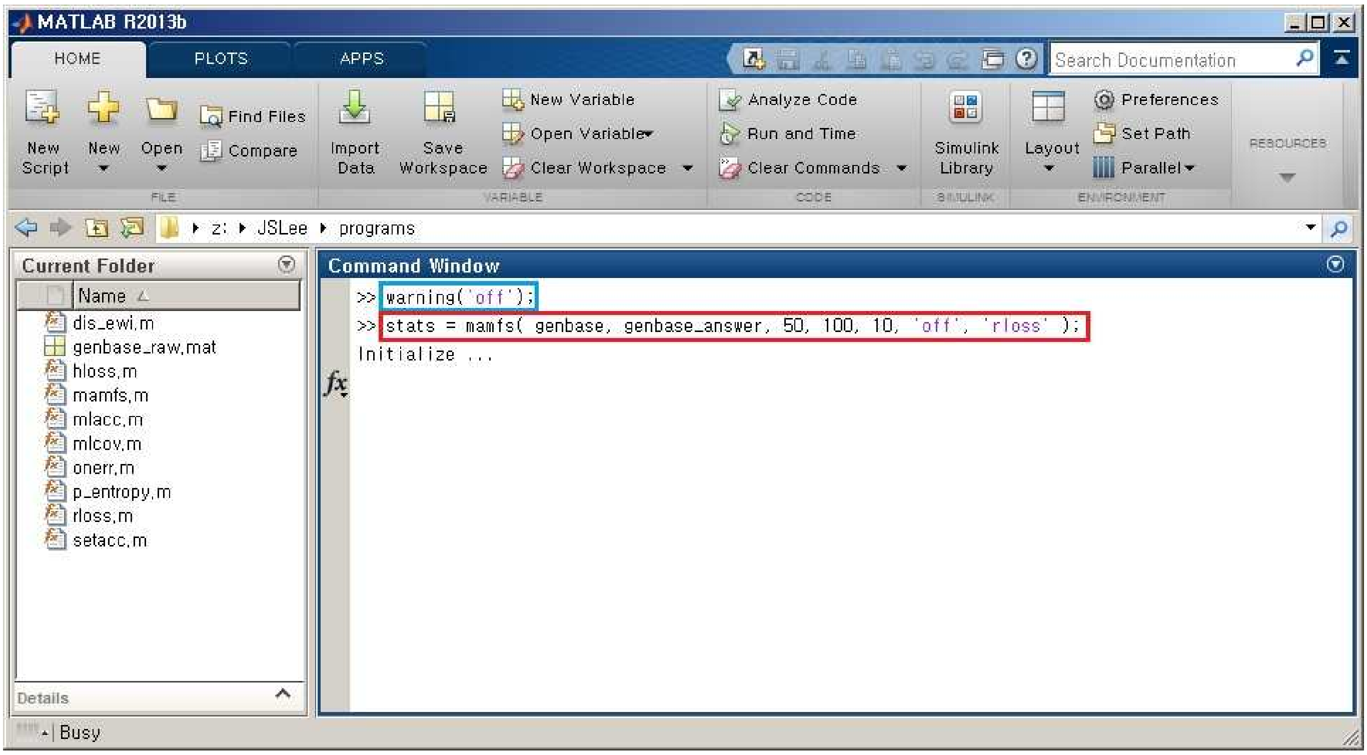
Note

A. To load the example file, type "load genbase_raw.m" to "Command Window".

Check points

A. If the example file is loaded successfully, you will see four variables (genbase, genbase_answer, genbase_lp_answer, and sim_seq) from "Workspace".

Step 4. Execute Memetic Algorithm for Multi-label Feature Selection (MAMFS).



```
>> warning('off');
>> stats = mamfs( genbase, genbase_answer, 50, 100, 10, 'off', 'rloss' );
Initialize ...
```

Note

A. "warning('off');" function turns off MATLAB global warning messages. For clarity, we recommend to run this option before actual execution of MAMFS.

B. To execute MAMFS, type "stats = mamfs(genbase, genbase_answer, 50, 100, 10, 'off', 'rloss');" to "Command Window".

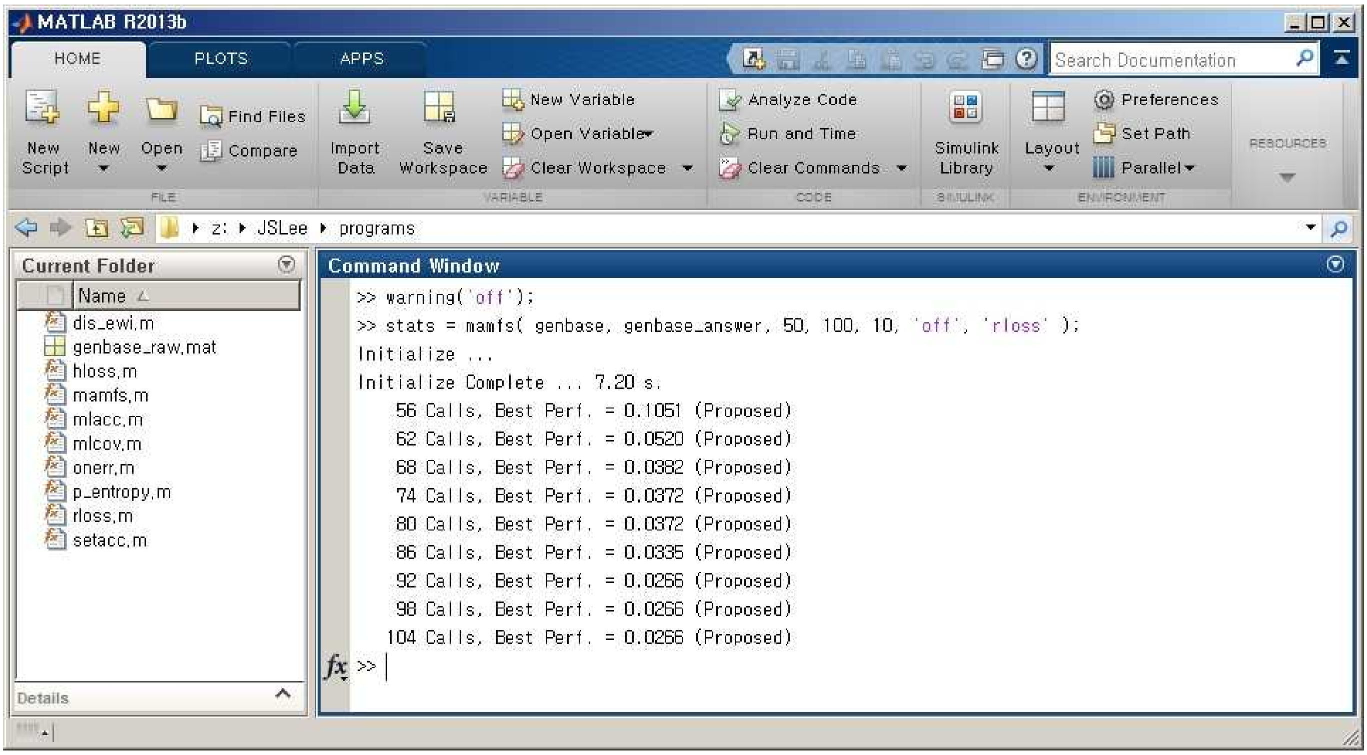
Check points

A. If the MAMFS is run correctly, you will see "Initialize ..." message from "Command Window".

B. Detailed information about each variable is given below.

Variable	Explanation
stats	1×3 output cell matrix. Detailed information will be given in Step 6.
genbase	The data set matrix that is composed of 662 patterns and 1,185 features.
genbase_answer	The ground truth matrix that is composed of 27 labels (See Step 3).
50	The size of population (or the number of chromosomes in the population).
100	The maximum number of allowed fitness function calls. If MAMFS spends 100 fitness function calls during its execution, it will be terminated and returns "stats" variable.
10	The maximum size (or cardinality) of selected features. The size of feature subset will be smaller than specified value.
'off'	If the input data matrix is a categorical (or binary) data set, it should be set to 'off'. In contrast, it should be set to 'on' if the input data matrix is a numerical data set.
'rloss'	The name of fitness function. MAMFS allows six types of evaluation measures. ● 'hloss': Hamming loss ● 'rloss': Ranking loss ● 'mlacc': Multi-label accuracy ● 'setacc': Set accuracy ● 'onerr': One error ● 'mlcov': Coverage

Step 5. Wait a second until MAMFS returns the output variable.



```
>> warning('off');
>> stats = mamfs( genbase, genbase_answer, 50, 100, 10, 'off', 'rloss' );
Initialize ...
Initialize Complete ... 7.20 s.
    56 Calls, Best Perf. = 0.1051 (Proposed)
    62 Calls, Best Perf. = 0.0520 (Proposed)
    68 Calls, Best Perf. = 0.0382 (Proposed)
    74 Calls, Best Perf. = 0.0372 (Proposed)
    80 Calls, Best Perf. = 0.0372 (Proposed)
    86 Calls, Best Perf. = 0.0335 (Proposed)
    92 Calls, Best Perf. = 0.0266 (Proposed)
    98 Calls, Best Perf. = 0.0266 (Proposed)
   104 Calls, Best Perf. = 0.0266 (Proposed)
fx >> |
```

Note

A. Each line shows the fitness value of the best chromosome. For example, in this tutorial, MAMFS found a chromosome that gave 0.0266 of Ranking loss value after spending 104 fitness function calls (FFCs).

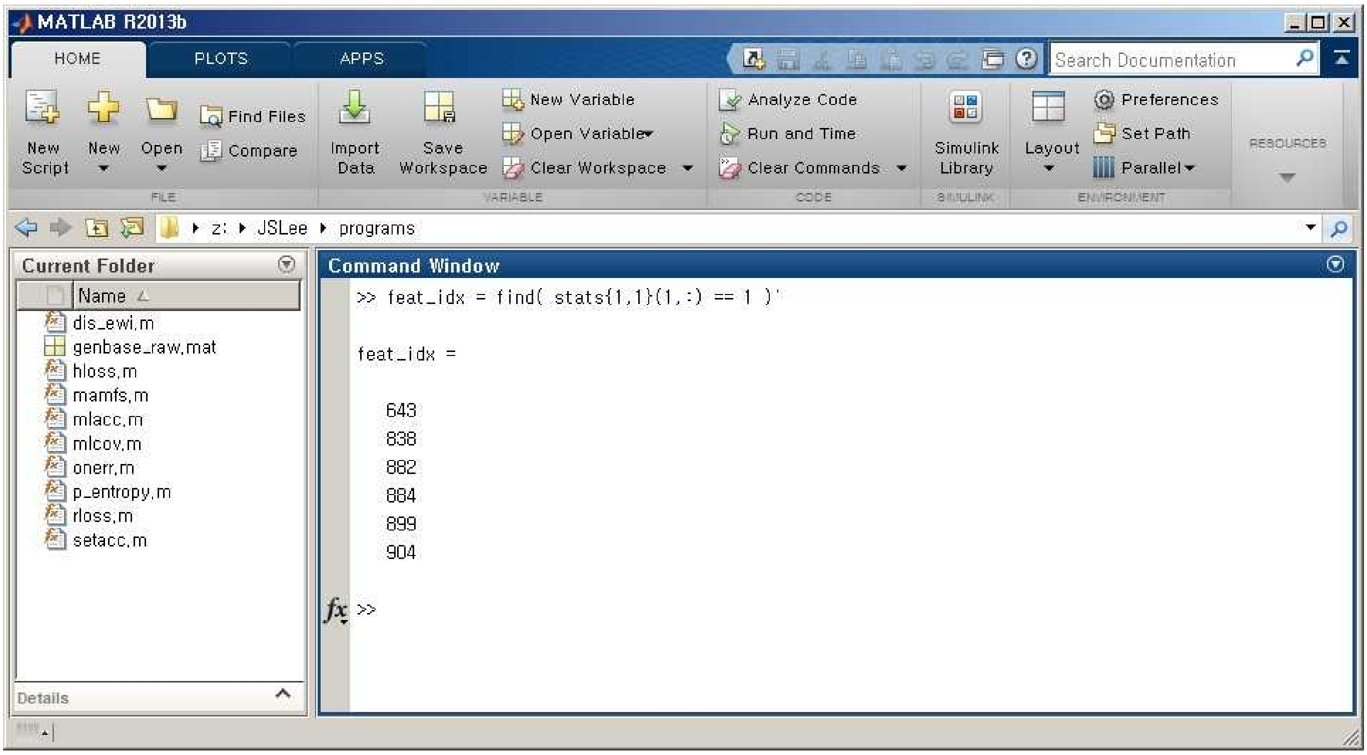
Check points

A. MAMFS displays the best fitness value for each 6 FFCs. Specific fitness value for each FFCs can be easily obtained as below:

Fitness Function Calls (FFCs)	Fitness (Ranking loss)
56-61	0.1051
62-67	0.0520
68-73	0.0382
74-79	0.0372
80-85	0.0372
86-91	0.0335
92-97	0.0266
98-103	0.0266
104 (Final)	0.0266

B. The gap of 6 FFCs comes from the genetic and local search process of MAMFS; 3 FFCs for local search and 3 FFCs for genetic search. During the creation or refinement process, MAMFS does not update the population, but spends FFCs for evaluating new chromosomes (or offspring).

Step 6. Get the index of selected features.



```
>> feat_idx = find( stats{1,1}(1,:) == 1 )'

feat_idx =

   643
   838
   882
   884
   899
   904

fx >>
```

Note

A. After MAMFS returns "stats" variable, you can obtain the specific index of selected feature. To obtain the index, type "feat_idx = find(stats{1,1}(1,:) == 1)'" to "Command Window".

B. In this tutorial, MAMFS returns a feature subset that is composed of 643th, 838th, 882th, 884th, 899th, 904th features in "genbase" data set.

Check points

A. The "stats" variable is composed of three sub-cells. The meaning of each cell is given below:

<table>
<tr><th>Position</th><th colspan="2">Explanation</th></tr>
<tr><td>1st</td><td colspan="2">50×1,185 matrix that represents the final population of MAMFS. It is sorted based on the given fitness value (Best to Worst). The values 0/1 represent selected/discarded features.</td></tr>
<tr><td>2nd</td><td colspan="2">The final fitness value of MAMFS.</td></tr>
<tr><td rowspan="4">3rd</td><td colspan="2">The 3rd cell is composed of five sub-cells. Meaning of each cell is given below:</td></tr>
<tr><td>1st</td><td>The number of spent FFCs</td></tr>
<tr><td>2nd</td><td>The fitness value of corresponding FFCs described in the 1st cell.</td></tr>
<tr><td>Remaining</td><td>Detailed fitness values correspond to given evaluation measure. If there is no specific statistics, it will be set to 'NaN' value.</td></tr>
</table>