=====================================================================

0. Introduction

=====================================================================

Feature Extraction & Feature Selection

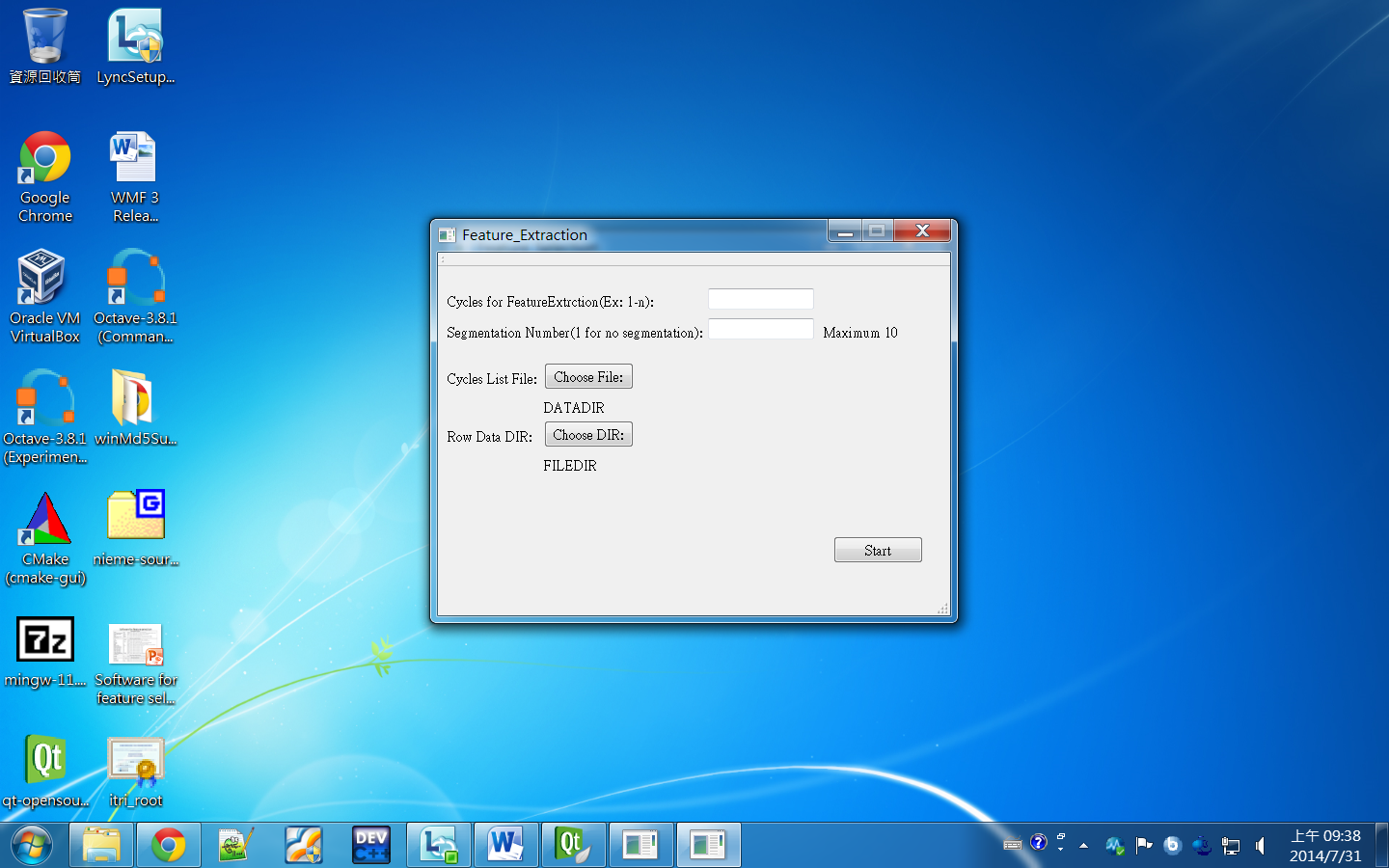
=====================================================================

1. Program Usage

=====================================================================

GUI 版本：

Feature\_Extraction：



Cycles for FeatureExtraction (Ex: 1-n)：欲使用的Cycle數目，如第一個cycle輸入1-1，1~5 cycles輸入1-5，以此類推。

Segmentation Number(1 for no segmentation)：在每個run裡面進行切分動作再計算feature，1 表示不切分，最大值為10。

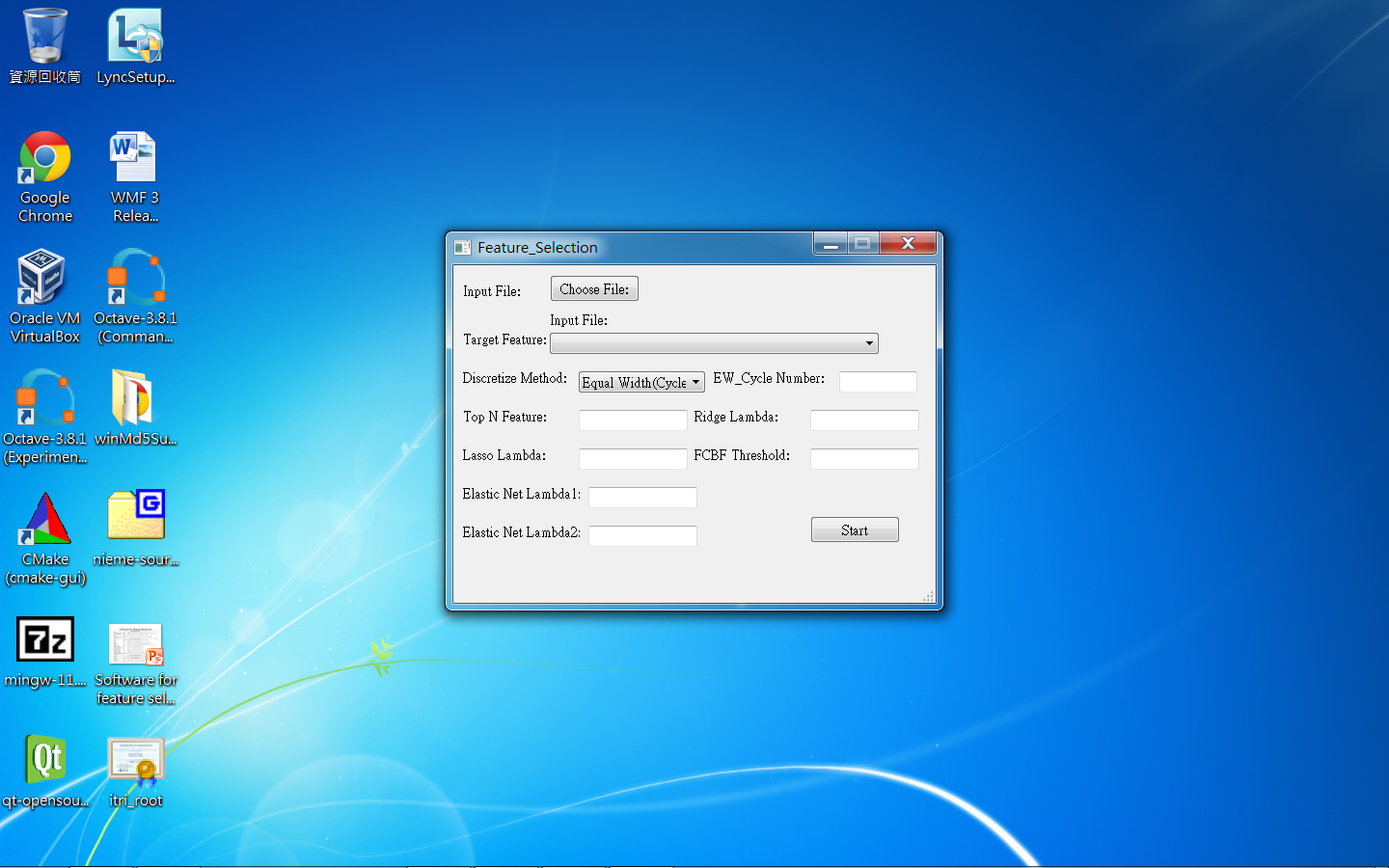
Cycle List File：點選Choose File按鈕，選取use\_file\_list.csv檔案所在位置，選取後路徑將顯示於下方。

Row Data DIR：點選Choose File按鈕，選取ROW Data所在資料夾，選取後路徑將顯示於下方。

點選Start後將執行，成功出現”Finish”字樣的對話框，失敗則為”Fail”。

Output 為一至兩個csv file，如為no segmentation output 1 file，如有segmentation output兩個file，分別為切分後兩種擺法(增加資料量(直擺)，增加資料維度(橫擺))。

Feature\_Selection：



Input File：點擊Choose File:按鈕，選取Feature\_Extraction所輸出的檔案，檔案名稱將於選取後顯示於下方。

Target Feature：自動讀取檔案內Feature名稱，可挑選其一作為分析的目標值。(於此數據中應是挑選DP\_Filter相關數值)

Discretize Method：

1. Equal Width(Cycle) 等距切割進行數值離散化，於後方輸入離散化數值數目。
2. Mannual 手動指定離散化切割點。(EX：5,10,15)

Top N Features：輸入n，輸出將會輸出前N項選出來的Feature。(註：某些演算法選出來的Feature可能只有少數個，不一定有N個)

Ridge Lambda：進行Ridge迴歸的Lambda值，1-3。

Lasso Lambda：進行Lasso迴歸的Lambda值，1-3。

Elastic Net Lambda 1：進行Elastic Net迴歸的Lambda1值，1-3。

Elastic Net Lambda 2：進行Elastic Net迴歸的Lambda2值，1-3。

點選Start後將執行，成功出現”Finish”字樣的對話框，失敗則為”Fail”。

BASH版本：

FE\_no\_GUI.exe dir(cyclelist.csv) dir(rowdata) Cyclebegin(num) Cycleend(num) segmentnum

dir(cyclelist.csv) 絕對相對路徑皆可

dir(rowdata) 絕對相對路徑皆可

Cyclebegin(num) 起始cycle number

Cycleend(num) 結束cycle number

segmentnum 等份數

共5個參數

FS\_no\_GUI.exe input\_file target\_feature top\_k disct\_method fcbf\_thrd ridge\_lambda lasso\_lambda els\_lambda1 els\_lambda2

input\_file FE處理後的檔案(Output\_noSeg.csv, Output\_seg1.csv, Output\_seg2.csv)

target\_feature 目標特徵("dP\_Filter (X1)\_max")

top\_k 演算法篩選的個數(15)

disct\_method 指定離散化方式("ew\_cycle=4", "5,15,20")

fcbf\_thrd MI-FCBF演算法之threshold (0.01)

ridge\_lambda Regression-RIDGE演算法之lambda(1, 2, 3)

lasso\_lambda Regression-LASSO演算法之lambda(1, 2, 3)

els\_lambda1 Regression-ElasticNet演算法之lambda1(1, 2, 3)

els\_lambda2 Regression-ElasticNet演算法之lambda2(1, 2, 3)

共9個參數

=====================================================================

2. Build from Soucre Code

=====================================================================

FeatureExtraction:

1.gsl 1.15 32bit https://code.google.com/p/oscats/downloads/list

說明：解壓放C根目錄 Compile 時 linker 下指令連結此函式庫

EX: ../../../../GSL-1.15/lib/libgsl.a ../../../../GSL-1.15/lib/libgslcblas.a

2.QT 設計gui

32bit 5.3 mingw 4.8.2

default dll linking

如果要standalone的exe,需built static qt

附上 built 好的static版 如無法使用需重built 參考以下

http://qt-project.org/wiki/How-to-build-a-static-Qt-for-Windows-MinGW

3.DataBase:

如需使用快速讀取功能(csv.h),compile時加以下指令

-std=c++11 -D\_\_NO\_MINGW\_LFS -DUSE\_FAST\_CSV

Fast CSV can only read fixed CSV format (i.e. exactly 8 columns containing "DataTime", "dP\_Filter (X1)", ...)

FeatureSelecyion:

Regression based:

mlpack(使用mingw make)

1.libxml libconv

下載位置(直接使用prebuilt的只能使用.dll.a版本)

ftp://ftp.zlatkovic.com/pub/libxml/64bit/

或是自己重編static版(我是自己編)

http://stackoverflow.com/questions/3429101/building-the-latest-iconv-and-libxml2-binaries-in-win32

linker指令(如果需要的話)

..\libxml2-2.9.1-win32-x86\lib\libxml2.dll.a ..\libxml2-2.9.1-win32-x86\lib\libxml2.a ..\libiconv-1.8-20020830\lib\libiconv.a ..\libiconv-1.8-20020830\lib\libcharset.a

2.armadillo-4.320.0(enable lapack/blas)

如何enable lapack/blas：

include\armadillo\_bits\config.hpp

#define ARMA\_USE\_LAPACK

#define ARMA\_USE\_BLAS

這兩行uncomment

linker指令

..\Lapack\_win32\_release\blas\_win32\_MT.lib ..\Lapack\_win32\_release\lapack\_win32\_MT.lib

並把兩dll放到執行檔旁

3.何處下載lapack/blas

http://ylzhao.blogspot.tw/2013/10/blas-lapack-precompiled-binaries-for.html

4.boost

可以自己編或使用含boost的mingw

含boost mingw :http://nuwen.net/mingw.html

自己編:

http://stackoverflow.com/questions/20265879/how-to-build-boost-1-55-with-mingw

http://www.boost.org/

4.mlpack本身(很難編...)

參考

http://www.mlpack.org/trac/wiki/MLPACKOnWindows

所需套件下載完後(lapack 64bit,libxml2 64bit)

注意Cmakelists裡面comment掉下面這兩段段再用cmake:

#if (WIN32)

# link\_directories(${Boost\_LIBRARY\_DIRS})

# set(Boost\_LIBRARIES "")

#endif (WIN32)

#add\_definitions(-DBOOST\_TEST\_DYN\_LINK)

cmake裡可加上下列兩個entry

Boost\_NO\_BOOST\_CMAKE

Boost\_USE\_STATIC\_LIBS

另外，如果compile之後錯在tree\_test.cpp

comment 掉log::的部分(共有兩行)

MI based:

FEAST:

FEAST資料夾內

Makefile做下列更改

libFSToolbox.so : $(objects)

$(LINKER) -L$(MITOOLBOXPATH) libMIToolbox.so -lm -shared -o libFSToolbox.so $(objects)

先compile MIToolbox

再compile FEAST