Using matlab in Java

Method 1: JMI

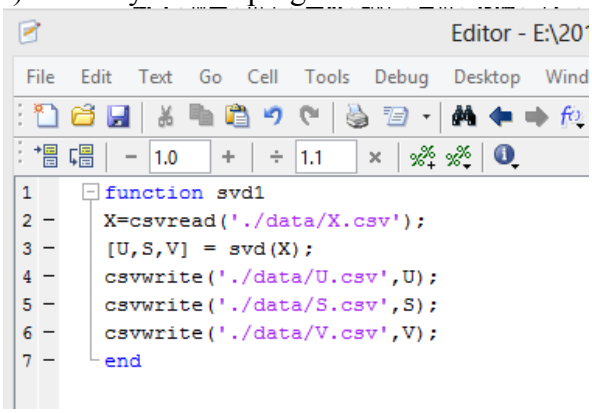
Check http://code.google.com/p/matlabcontrol/

1. **Download JMI package and include it in build path.**
2. **Follow the instructions to call matlab in your program.**

Example of SVD, PCA and LDA:

SVD:

a). Write your svd program in matlab:



b). call matlab in java

```
import matlabcontrol.*;

public class test {

          /**
          * @param args
          */
      public static void main(String[] args) throws
MatlabConnectionException, MatlabInvocationException
      {
          //Create a proxy, which we will use to control MATLAB
          MatlabProxyFactory factory = new MatlabProxyFactory();
          MatlabProxy proxy = factory.getProxy();
          //set matlab path
          String path = "cd(\'*******\')";
          proxy.eval(path);
          //call svd
          proxy.eval("svd1");

          //Disconnect the proxy from MATLAB
          proxy.disconnect();
      }
}
```

PCA

Similar with svd call matlab building function [COEFF,SCORE] = princomp(X) for your program

LDA

Using Matlab Topic Modeling Toolbox 1.4

http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

Adjust LDA(focus on exampleLDA1) for your own program, including parameters setting etc.

Please refer to svd example for the following steps.

Method 2: Builder JA

1. Environment setting

a) Make sure you have Eclipse and Matlab installed on your machine.

b) Environment variables setting

i. JAVA_HOME (e.f. C: \Java)

ii. Classpath

adding *matlabInstallRoot*\toolbox\javabuilder\jar\javabuilder.jar

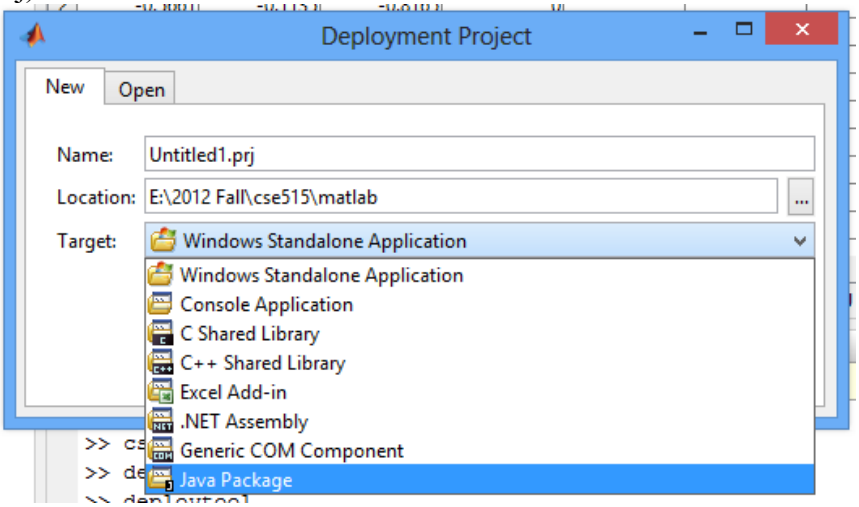
iii. Path

adding %JAVA_HOME%/bin/javac

2. build matlab m-file into a jar

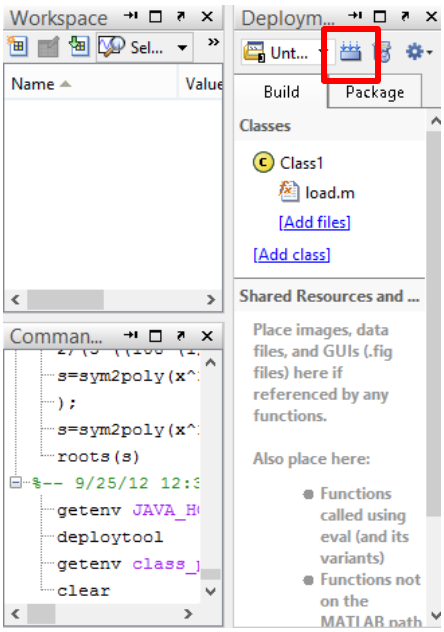
a) type deploytool in matlab command window. Deployment Tool will pop out.

b) In Deployment Tool click new, choose Java Package and project name (e.f. test.prj).

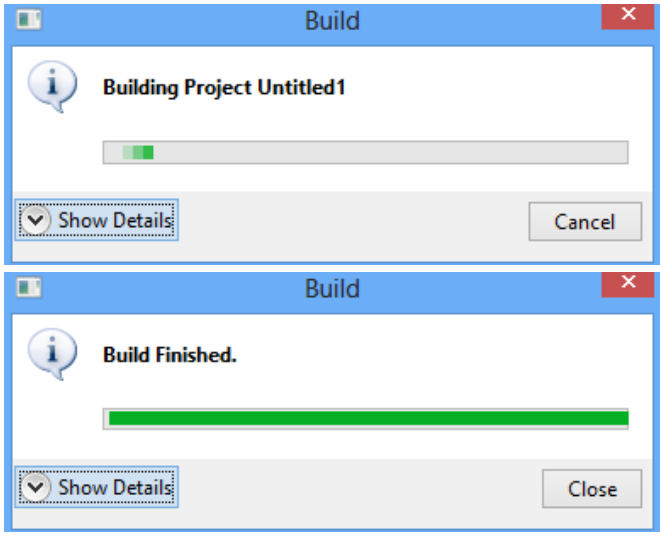


c) In the Deployment Tool pane, ensure that the Generate Verbose Output option is selected

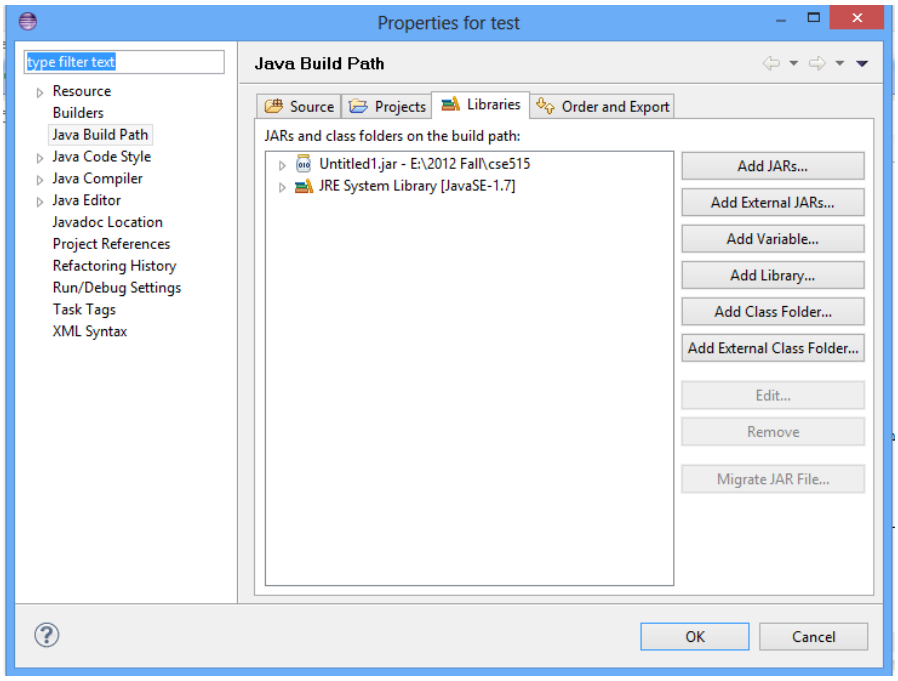
d) Drag .m file into Deployment Tool's class folder.



e) Click build button



3. Add newly build .jar into your own program.



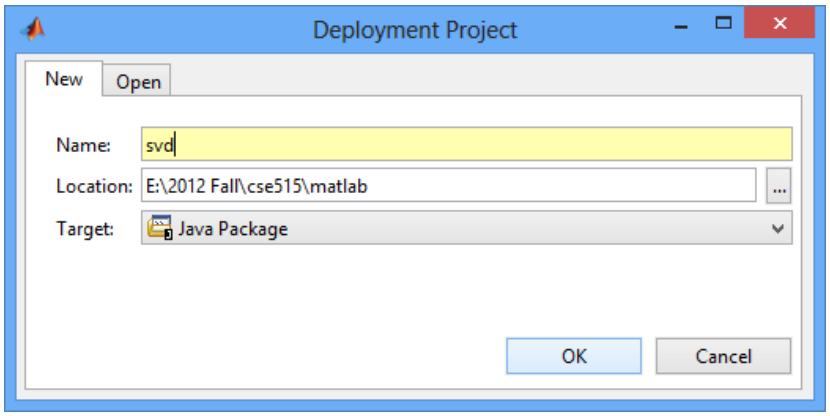
Note. please also add javabuilder.jar to external libs.
(can be find at: *:\Program Files\MATLAB\R2010a\toolbox\javabuilder\jar)

Finish!

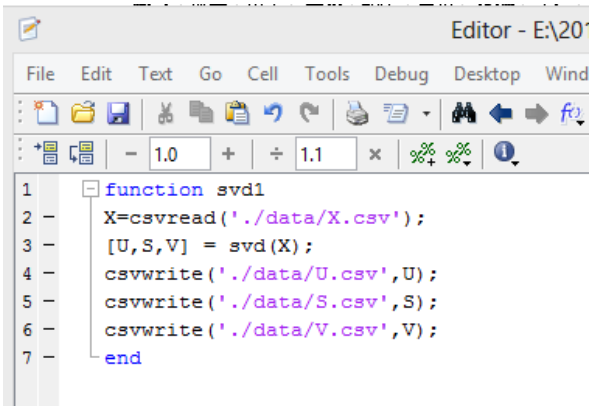
Example of SVD, PCA and LDA:

SVD

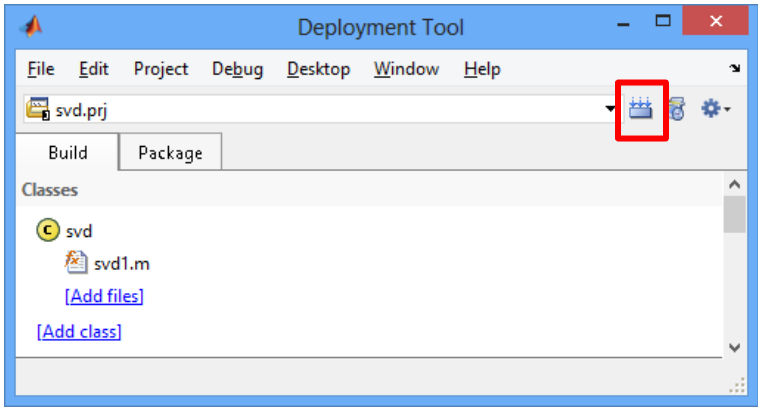
a). create Java package project (enter deploytool in command window)



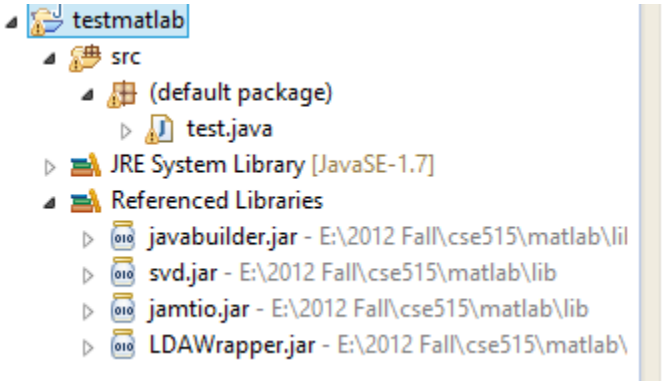
b). write .m file in matlab that call matlab svd function.



c). save your .m file and drag your function into java package class
Build java package



d). include your matlab package in your java project.



e). write your program.

```
import com.mathworks.toolbox.javabuilder.*;
import svd.*;

public class test {

        /**
        * @param args
```

```
        */
        public static void main(String[] args) {
                try {
                svd svdclass=new svd();
                System.out.println("svd");
                svdclass.svd1();
                }catch (Exception e){System.out.println(e);}
        }
}
```

Finish.

PCA

Similar with svd call matlab building function [COEFF,SCORE] = princomp(X) for your program

LDA

Using Matlab Topic Modeling Toolbox 1.4

http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

Adjust LDA(focus on exampleLDA1) for your own program, including parameters setting etc.

Drag all functions that you need to use to new java package project in matlab.

Build java package.

Please refer to svd example for the following steps.

Note:

You may interested in using JMatIO(Matlab's MAT-file I/O in JAVA) in your project.

check http://www.mathworks.com/matlabcentral/fileexchange/10759 for details.