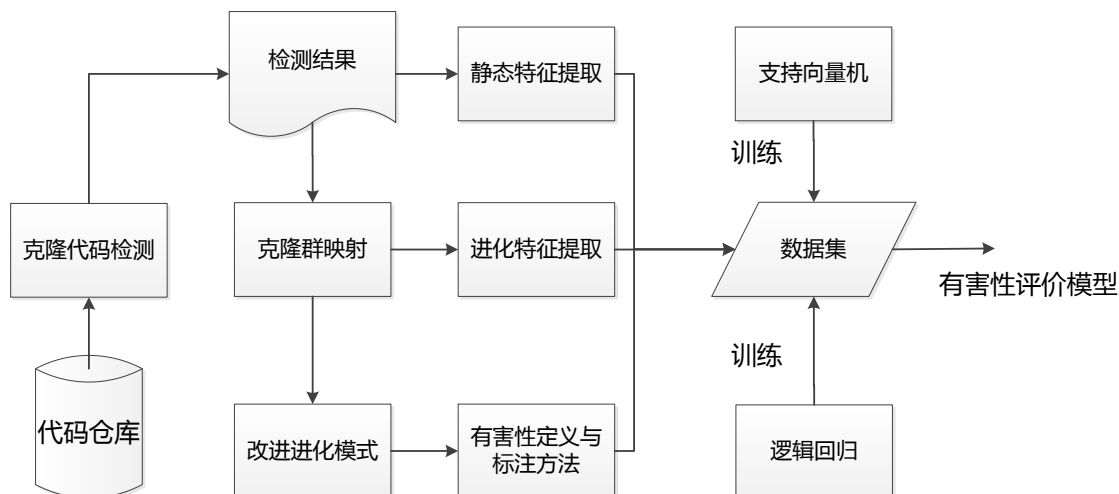


基于 SVM 的克隆代码有害性评价方法说明

整体研究框架如图所示，



使用 SVM 对克隆代码进行有害性评价的详细工作过程如下：首先通过版本控制系统 Subversion 或 sourceforge.net 等网站获取实验数据——连续多个版本的开源软件源代码。然后使用 NiCad 工具对这些源代码执行克隆代码检测，克隆代码检测的结果有两方面用途：一方面直接用来做静态特征提取，另一方面用来进行克隆群映射。克隆群映射的结果也有两方面用途：一方面用来做进化特征提取，另一方面用来对样本有害性作标注。如此经过如上两步的处理，有害性评价模型所需的样本即准备完成。接着，将样本划分为训练集与测试集，使用 SVM 模型对其进行训练，经过交叉验证、调整参数等过程之后，一个初步的基于 SVM 的克隆代码有害性评价模型即建立完成，上图的输出对于本文选用的实现方式而言是 `evaluation.model` 文件，利用它可以对某段未知有害性的克隆代码做出评价。另外，如图所示，为了对 SVM 模型的性能进行比较，在训练样本模块又加入了逻辑回归模型。

实验数据获取

sourceforge.net 等网站

真是一个伟大的网站：<http://www.souceforge.net>

还有 <http://www.oldlinux.org> 等

sourceforge

Search

BrowseEnterpriseBlogHelpJobs

SOLUTION CENTERSGo ParallelSmarter ITNewsletters

[Home](#) / [Browse](#) / [Name Service \(DNS\)](#) / [dnsjava](#) / [Support](#)

dnsjava

Brought to you by: [bwellington](#)

SummaryFilesReviewsSupportWikiMailing ListsTicketsNewsDiscussionCode

Looking for the latest version? [Download dnsjava-2.1.5.jar \(301.2 kB\)](#)

[Home](#) / [dnsjava](#)

Name	Modified	Size	Downloads
↑ Parent folder			
2.1.5	2013-04-10		
2.1.4	2013-01-04		
2.1.3	2011-10-24		
2.1.2	2011-07-25		
2.1.1	2011-02-10		
2.1.0	2010-09-08		
2.0.8	2009-11-21		

TortoiseSVN + SVNKit

TortoiseSVN 为主流的版本控制系统 SVN 的 windows 客户端，可以提取版本库；SVNKit 为 SVN 的纯 java 客户端库，通过对 SVN 进行二次开发，可以提取版本日志信息。

CVSNT + TortoiseCVS

CVSNT 为版本控制系统 CVS 的服务器端，TortoiseCVS 为 CVS 的客户端；CVSNT 为付费软件，难以破解，且存在各种兼容问题，有待解决，作为 SVN 的候选。

克隆代码检测

详见文档《总结 NICAD 工具的安装过程》。

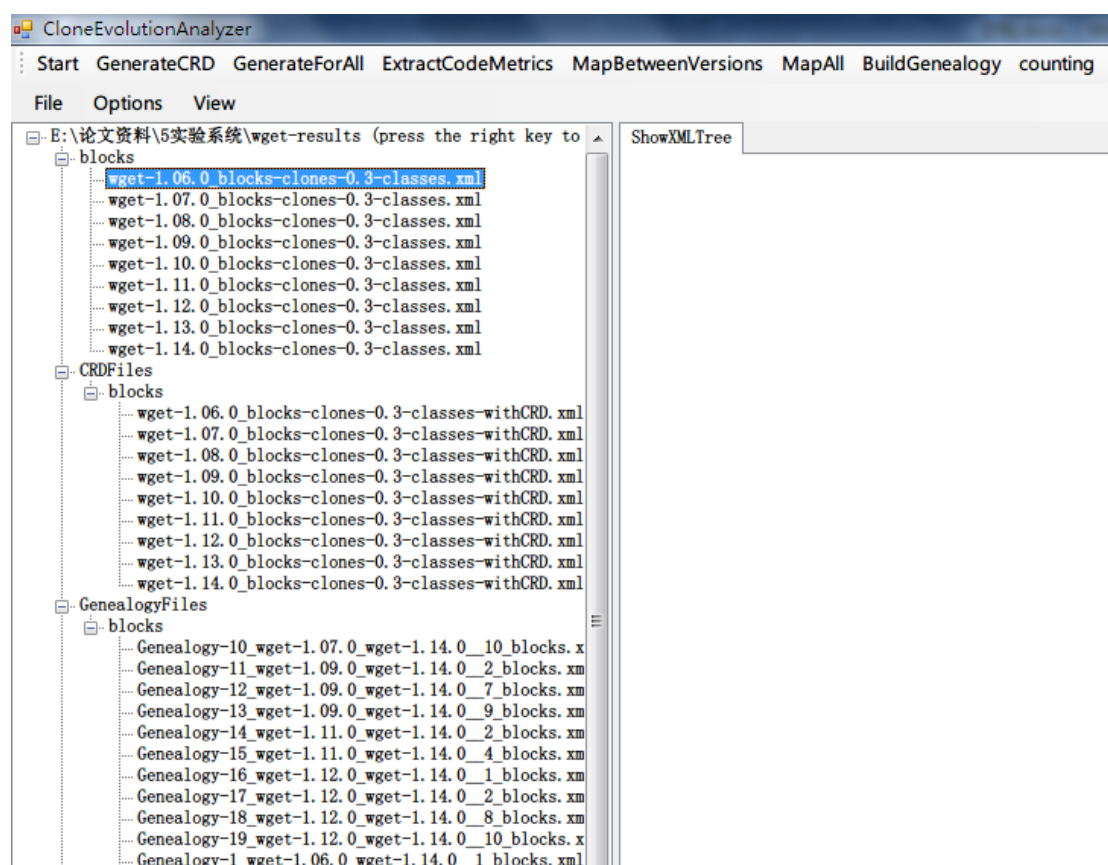
值得注意的是：

- （1）需要熟悉一些常见 Linux 命令：cat, su, chmod, mv 等
- （2）命令头是 nicad3

- (3) 注意命令中的路径
- (4) 级别 blocks, 相似度阈值 0.7 (默认)
- (5) C#检测结果文件名有 bug

克隆群映射





使用慈萌同学开发的工具 CGE。



先“GenerateForAll”生成 CRD 文件

再 “MapAll”生成 MAP 文件

得到如下的文件以备后用

 blocks	2013/5/19 15:11	文件夹
 CRDFiles	2013/5/19 15:53	文件夹
 GenealogyFiles	2013/5/19 16:07	文件夹
 MAPFiles	2013/5/19 16:07	文件夹

特征提取

在 CGE 基础上加了提取 Halstead 度量的类, 在提取 Halstead 度量前还要在

Preprocess 类中加入删除字符串的函数。

数据预处理

先转换 xml 文件的格式，否则 python 的 minidom 模块处理不了：

C:\Users\founder\utf8.py E:\wget-results\CRDFiles\blocks\

然后提取特征：

C:\Users\founder\extract.py

E:\wget-results\CRDFiles\blocks\

E:\wget-results\MAPFiles\blocks\

如有必要，进行欠采样：

undersample.py [file]

LibSVM+LibLinear 建模

1.Libsvm 下载 <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Gnuplot 下载: <http://www.gnuplot.info/>

Python 下载: <http://www.python.org/getit/>

这里我下的 libsvm 版本为 3.1.2,gnuplot 版本为 4.6.0, python 版本为 2.7.3。

其中 libsvm 的作用就不用多介绍了，gnuplot 是图像绘画工具，可以将数据可视化，python 是一种程序编程语言，很方便，所以 libsvm 和她走得比较近。

我将 libsvm 解压(即相当于安装)在 C:\Program Files\libsvm-3.12 下

Gnuplot 安装在 C:\Program Files\gnuplot 下

Python 安装在 C:\Program Files\Python27 下

2.设置 python 的路径(我的电脑->右键->属性->高级->环境变量->系统变量->path)。

【注】在修改完路径变量后一定要重启机器，否则程序仍不能正常运行。

3.修改代码：

(1) 修改 easy.py 中的代码：

example for windows 下面有一些需要的文件相对当前文件 easy.py 的路径(即相对路径)

注：这里 r 是 raw 的缩写，也可以用 R，表示后面字符串中的“\”不作为转义字符。

①gnuplot_exe = r"..\\gnuplot\\bin\\pgnuplot.exe" (因为前面已经将 gnuplot 目录作为 LIBSVM 的一个子目录了)

②grid_py = r"..\\grid.py"

③cmd = 'python %s -svmtrain "%s" -gnuplot "%s" "%s"' % (grid_py, svmtrain_exe, gnuplot_exe, scaled_file)

(2) 修改 grid.py 中的代码:

```
gnuplot_exe = r"..\\gnuplot\\bin\\gnuplot.exe"
```

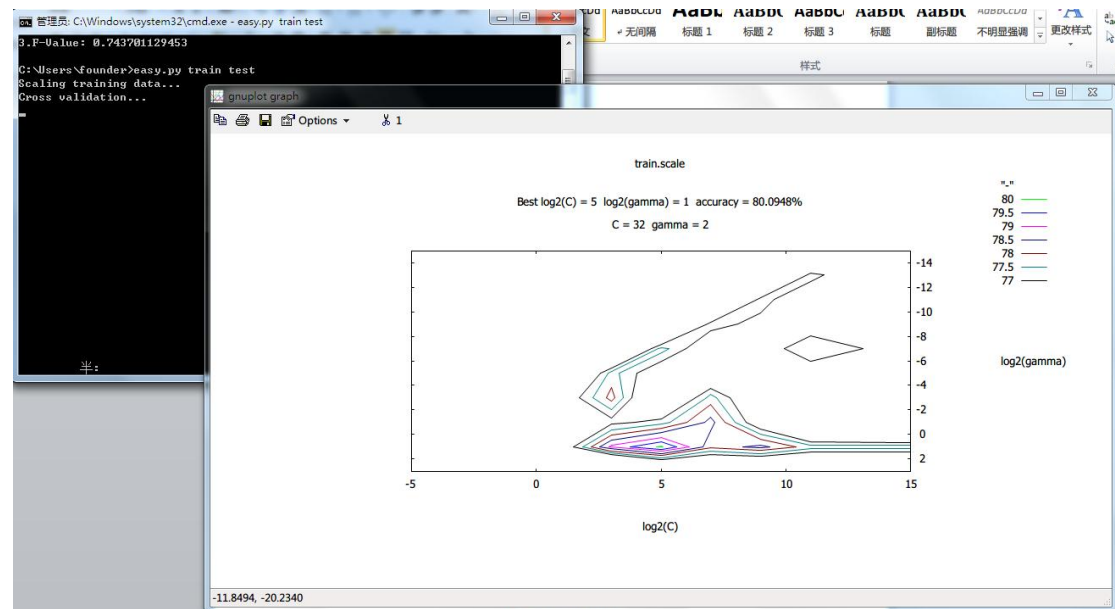
4.检查是否成功:

首先, 运行 cmd, 将当前目录设为 D:\Setup Files\LIBSVM 2.89\tools, 也即 easy.py 所在的目录。因为 easy.py 中的目录都是以该目录为标准的相对目录。

在 dos 提示符后运行:python easy.py heart_scale, 如果弹出一个 gnuplot 界面, 则表示配置成功。其中, heart_scale 为 tools 目录下的一个特征数据文件, 专门用来练习、测试用的。

命令行:

```
easy.py [train_file] [test_file]
```



统计实验结果

result+.py

一个典型的结果如图所示:

```

Best c=128.0, g=2.0 CV rate=93.4197
Training...
Output model: train.model
Scaling testing data...
Testing...
Accuracy = 71.63% (1270/1773) (classification)
Output prediction: test.predict

C:\Users\founder>result.py
Result:
+++++
1.# of + in train: 970
2.# of - in train: 3893
+++++
1.# of + in test: 532
2.# of - in test: 1241
+++++
1.Precision: 0.5185659411011524
2.Recall: 0.7612781954887218
3.F-Value: 0.6169078446306169

```

注：信息增益的结果在全部属性组提取时给出

附：各程序说明

utf8.py	转换 xml 文件的格式（从 GB2312 到 UTF-8）
extract.py	提取全部特征组的样本
extract_without_content.py	提取无内容组的样本
extract_without_evolution.py	提取无进化组的样本
extract_without_size.py	提取无容量组的样本
result.py	显示实验结果
result+.py	显示实验结果及正负例均衡判断
undersample.py	欠采样
guitest.py	界面