Part I: Subset selection tools  
  
Introduction  
============  
  
Training large data is time consuming. Sometimes one should work on a  
smaller subset first. The python script subset.py randomly selects a  
specified number of samples. For classification data, we provide a  
stratified selection to ensure the same class distribution in the  
subset.  
  
Usage: subset.py [options] dataset number [output1] [output2]  
  
This script selects a subset of the given data set.  
  
options:  
-s method : method of selection (default 0)  
 0 -- stratified selection (classification only)  
 1 -- random selection  
  
output1 : the subset (optional)  
output2 : the rest of data (optional)  
  
If output1 is omitted, the subset will be printed on the screen.  
  
Example  
=======  
  
> python subset.py heart\_scale 100 file1 file2  
  
From heart\_scale 100 samples are randomly selected and stored in  
file1. All remaining instances are stored in file2.  
  
  
Part II: Parameter Selection Tools  
  
Introduction  
============  
  
grid.py is a parameter selection tool for C-SVM classification using  
the RBF (radial basis function) kernel. It uses cross validation (CV)  
technique to estimate the accuracy of each parameter combination in  
the specified range and helps you to decide the best parameters for  
your problem.  
  
grid.py directly executes libsvm binaries (so no python binding is needed)  
for cross validation and then draw contour of CV accuracy using gnuplot.  
You must have libsvm and gnuplot installed before using it. The package  
gnuplot is available at<http://www.gnuplot.info/>  
  
On Mac OSX, the precompiled gnuplot file needs the library Aquarterm,  
which thus must be installed as well. In addition, this version of  
gnuplot does not support png, so you need to change "set term png  
transparent small" and use other image formats. For example, you may  
have "set term pbm small color".  
  
Usage: grid.py [grid\_options] [svm\_options] dataset  
  
grid\_options :  
-log2c {begin,end,step | "null"} : set the range of c (default -5,15,2)  
 begin,end,step -- c\_range = 2^{begin,...,begin+k\*step,...,end}  
 "null" -- do not grid with c  
-log2g {begin,end,step | "null"} : set the range of g (default 3,-15,-2)  
 begin,end,step -- g\_range = 2^{begin,...,begin+k\*step,...,end}  
 "null" -- do not grid with g  
-v n : n-fold cross validation (default 5)  
-svmtrain pathname : set svm executable path and name  
-gnuplot {pathname | "null"} :  
 pathname -- set gnuplot executable path and name  
 "null" -- do not plot   
-out {pathname | "null"} : (default dataset.out)  
 pathname -- set output file path and name  
 "null" -- do not output file  
-png pathname : set graphic output file path and name (default dataset.png)  
-resume [pathname] : resume the grid task using an existing output file (default pathname is dataset.out)  
 Use this option only if some parameters have been checked for the SAME data.  
  
svm\_options : additional options for svm-train  
  
The program conducts v-fold cross validation using parameter C (and gamma)  
= 2^begin, 2^(begin+step), ..., 2^end.  
  
You can specify where the libsvm executable and gnuplot are using the  
-svmtrain and -gnuplot parameters.  
  
For windows users, please use pgnuplot.exe. If you are using gnuplot  
3.7.1, please upgrade to version 3.7.3 or higher. The version 3.7.1  
has a bug. If you use cygwin on windows, please use gunplot-x11.  
  
If the task is terminated accidentally or you would like to change the  
range of parameters, you can apply '-resume' to save time by re-using  
previous results. You may specify the output file of a previous run  
or use the default (i.e., dataset.out) without giving a name. Please  
note that the same condition must be used in two runs. For example,  
you cannot use '-v 10' earlier and resume the task with '-v 5'.  
  
The value of some options can be "null." For example, `-log2c -1,0,1  
-log2 "null"' means that C=2^-1,2^0,2^1 and g=LIBSVM's default gamma  
value. That is, you do not conduct parameter selection on gamma.  
  
Example  
=======  
  
> python grid.py -log2c -5,5,1 -log2g -4,0,1 -v 5 -m 300 heart\_scale  
  
Users (in particular MS Windows users) may need to specify the path of  
executable files. You can either change paths in the beginning of  
grid.py or specify them in the command line. For example,  
  
> grid.py -log2c -5,5,1 -svmtrain "c:\Program Files\libsvm\windows\svm-train.exe" -gnuplot c:\tmp\gnuplot\binary\pgnuplot.exe -v 10 heart\_scale  
  
Output: two files  
dataset.png: the CV accuracy contour plot generated by gnuplot  
dataset.out: the CV accuracy at each (log2(C),log2(gamma))  
  
The following example saves running time by loading the output file of a previous run.  
  
> python grid.py -log2c -7,7,1 -log2g -5,2,1 -v 5 -resume heart\_scale.out heart\_scale  
  
Parallel grid search  
====================  
  
You can conduct a parallel grid search by dispatching jobs to a  
cluster of computers which share the same file system. First, you add  
machine names in grid.py:  
  
ssh\_workers = ["linux1", "linux5", "linux5"]  
  
and then setup your ssh so that the authentication works without  
asking a password.  
  
The same machine (e.g., linux5 here) can be listed more than once if  
it has multiple CPUs or has more RAM. If the local machine is the  
best, you can also enlarge the nr\_local\_worker. For example:  
  
nr\_local\_worker = 2  
  
Example:  
  
> python grid.py heart\_scale  
[local] -1 -1 78.8889 (best c=0.5, g=0.5, rate=78.8889)  
[linux5] -1 -7 83.3333 (best c=0.5, g=0.0078125, rate=83.3333)  
[linux5] 5 -1 77.037 (best c=0.5, g=0.0078125, rate=83.3333)  
[linux1] 5 -7 83.3333 (best c=0.5, g=0.0078125, rate=83.3333)  
.  
.  
.  
  
If -log2c, -log2g, or -v is not specified, default values are used.  
  
If your system uses telnet instead of ssh, you list the computer names  
in telnet\_workers.  
  
Calling grid in Python  
======================  
  
In addition to using grid.py as a command-line tool, you can use it as a  
Python module.   
  
>>> rate, param = find\_parameters(dataset, options)  
  
You need to specify `dataset' and `options' (default ''). See the following example.  
  
> python  
  
>>> from grid import \*  
>>> rate, param = find\_parameters('../heart\_scale', '-log2c -1,1,1 -log2g -1,1,1')  
[local] 0.0 0.0 rate=74.8148 (best c=1.0, g=1.0, rate=74.8148)  
[local] 0.0 -1.0 rate=77.037 (best c=1.0, g=0.5, rate=77.037)  
.  
.  
[local] -1.0 -1.0 rate=78.8889 (best c=0.5, g=0.5, rate=78.8889)  
.  
.  
>>> rate  
78.8889  
>>> param  
{'c': 0.5, 'g': 0.5}  
  
  
Part III: LIBSVM format checking tools  
  
Introduction  
============  
  
`svm-train' conducts only a simple check of the input data. To do a  
detailed check, we provide a python script `checkdata.py.'  
  
Usage: checkdata.py dataset  
  
Exit status (returned value): 1 if there are errors, 0 otherwise.  
  
This tool is written by Rong-En Fan at National Taiwan University.  
  
Example  
=======  
  
> cat bad\_data  
1 3:1 2:4  
> python checkdata.py bad\_data  
line 1: feature indices must be in an ascending order, previous/current features 3:1 2:4  
Found 1 lines with error.