How To Run TAR3

Ying Hu
University of British Columbia
yingh@ece.ubc.ca

1 Getting TAR3

The TAR3 treatment learner is distributed under the GNU General Public License and is available online[1]. For installation, simply download the newest TAR3 package (dispatchTAR3.zip) and unzip it to your local computer. The whole package contains the following file structure:

- /bin: folder where all the executables reside
- /doc: related publications and user manual
- /sample: sample data sets and their configuration files
- /source: C source code

2 Configuration File

Table 1 lists the parameters used in the configuration file (xx.cfg). Note that the order of the parameters are not important, and if a parameter is missing, TAR3 will take the default value as listed in table 1.

TAR3 adopts a random sampling algorithm to draw treatments from the underlying distribution. Parameter `randomTrials` and `futileTrials` are part of the randomness control. Suppose we set:

```
maxNumber = 100
randomTrials = 50
futileTrials = 10
```

[1] http://www.ece.ubc.ca/twiki/bin/view/Softeng/TreatmentLearner

Name	Meaning	Default
granularity	how many intervals should a continuous attribute be divided	3
maxNumber	maximum number of treatments wanted	30
minSize	minimum treatment size expected in a single treatment	1
maxSize	maximum treatment size allowed in a single treatment	5
randomTrials	maximum random trials tried before stop	1
futileTrials	number of successive futile trials to be completed before stop	5
bestClass	percentage of best class examples expected to be remained in the treated set	50%

Table 1: Parameters seen in the configuration file.

In each random trial, TAR3 generates a set of treatments and maintains a list of 100 top ranked treatments. If a random trial doesn't contribute new treatments into that list (e.g., treatments generated in that trial have lower rank than those already in the list), it is called a futile trial. The process stops after completing 50 random trials or after 10 successive futile trials are reached. Empirically, setting `randomTrials` between 30 to 60 and `futileTrials` between 5 to 10 are usually sufficient to get stable treatments.

3 Name File

The `.names` file consists of a series of sections, each of which has restrictions and format. Blank lines, spaces, and tabs may be used to make the file more readable and have no significance. The vertical bar character(|) appearing anywhere on a line causes the rest of that line to be ignored, and can be used to incorporate comments in the file.

3.1 Name Restriction

1. A name cannot be the single character "?"

2. The special characters comma(,), colon(:), vertical bar(|), and backslash have particular meanings and must be escaped (preceded by a backslash character) if they appear in a name.
3. A period(.) may appear in a name provided it is not followed by a space.
4. Embedded spaces are also permitted in a name, but multiple white-space characters (spaces and tabs) are replaced by a single space.

3.2 Class Format

1. The first entry in the names file gives the class names, separated by commas.
2. There must be at least two class names.
3. Classes are ordered from the domain-specific point of view, with the worst first, the best last.

3.3 Attribute Format

1. An attribute entry begins with its name followed by a colon, and then a specification of the values it can take.
2. `continuous`: indicates that the attribute has numeric values, either integer or floating point.
3. A list of names separated by commas: indicates that the attribute has discrete values and specifies them explicitly. The order of attribute values is arbitrary.

3.4 Optional Sections

TAR2 takes the three sections as inputs to restrict the data processing scope of a particular data set.

- NOW section: NOW specifies the current status of the data, i.e., only those satisfy NOW criteria will be read in and processed. This data pre-process could always be obtained by using other tools.
- CHANGES section: CHANGES represents some desired zone within the data set that the user wishes to approach. Only attribute ranges specified in CHANGES could appear in the treatments.

- SCORE section: SCORE encodes user's preference of the classes. User can assign a specific score (weight) to a class. Without user specification, TAR3 scores the classes according to a default scoring function.

The above three sections are optional, but once they appear, their relative order is important. e.g., CHANGES section must be after NOW section and SCORE section must be the last.

3.5 Little Language

A little language is designed to specify attribute ranges in NOW and CHANGES sections, for example:

- `Attribute1:true`:
  all possible values are acceptable
- `Attribute2:ignore`:
  none values are acceptable
- `Attribute3:a, b, c`:
  for categorical attribute, only values a, b, c are acceptable
- `Attribute4:[-;10), [20;30], [50;-)`:
  for continuous attribute,the acceptable ranges are: $x < 10$ OR $20 <= x <= 30$ OR $x <= 50$

4 Command Line

Suppose the data set to use is `c:/tar3/data/myDataset.data`. The following files are required to be placed into the same folder:

- data file: `c:/tar3/data/myDataset.data`
- name file: `c:/tar3/data/myDataset.names`
- configuration file: `c:/tar3/data/myDataset.cfg`

To invoke TAR3, issue the command: (suppose `tar3.exe` resides in `c:/tar3/bin`)

```
cd c:/tar3/data
c:/tar3/bin/tar3 myDataset
```

5 Cross-Validation

TAR3 also comes with a cross-validation facility (`/bin/xval.exe`). This program is compatible with both TAR2(v2.2) and TAR3. To invoke it, use one of the following three commands:

- `xval tar2 fileName N`:
  N-way cross validation with tar2 on `fileName.data`
- `xval tar3 fileName N`:
  N-way cross validation with tar3 on `fileName.data`
- `xval -p  fileName N`:
  perform file-split on `fileName.data`

The current directory must be the one where the data files reside. If `tarX.exe` and `xval.exe` are in different folders from the current directory, full path must be specified. For example, to perform 10-way cross validation on `myDataset.data`, we issue the command:

```
cd c:/tar3/data
c:/tar3/bin/xval  c:/tar3/bin/tar3 myDataset 10
```

`xval.exe` first splits the data file in to N `.data` file and N `.test` files, resulting in:

```
XDF[0..N-1].data
XDF[0..N-1].test
```

Then it invokes tar2 or tar3 N times, generating N output files plus one summary file. After done, it automatically delete `XDF*.data` and `XDF*.test`