Nobody Knows What It's Like To Be the Bad Man
The Development Process for the `caret` Package

Max Kuhn, Ph.D

Pfizer Global R&D
Groton, CT
max.kuhn@pfizer.com

Outline

- What is the caret package?
- What makes it different
- Version control
- The CRAN release process
- Testing
- Documentation

Model Function Consistency

Since there are many modeling packages in R written by different people, there are some inconsistencies in how models are specified and predictions are created.

For example, many models have only one method of specifying the model (e.g. formula method only)

Generating Class Probabilities Using Different Packages

Function	`predict` Function Syntax
MASS:::lda	`predict(obj)` (no options needed)
stats:::glm	`predict(obj, type = "response")`
gbm:::gbm	`predict(obj, type = "response", n.trees)`
mda:::mda	`predict(obj, type = "posterior")`
rpart:::rpart	`predict(obj, type = "prob")`
RWeka:::Weka	`predict(obj, type = "probability")`
caTools:::LogitBoost	`predict(obj, type = "raw", nIter)`

The caret Package

The caret package was developed to:

- create a unified interface for modeling and prediction (interfaces to 180 models)
- streamline model tuning using resampling
- provide a variety of "helper" functions and classes for day–to–day model building tasks
- increase computational efficiency using parallel processing

First commits within Pfizer: 6/2005, First version on CRAN: 10/2007

Website: http://topepo.github.io/caret/

JSS Paper: http://www.jstatsoft.org/v28/i05/paper

Model List: http://topepo.github.io/caret/bytag.html

Many computing sections in APM

Package Dependencies

One thing that makes caret different from most other packages is that is uses code from an abnormally large number (> 80) of other packages.

Briefly, these were in the `Depends` field of the DESCRIPTION file which cause all of them to be loaded with caret.

For many years, they were moved to `Suggests`, which solved that issue,

However, their formal dependency in the DESCRIPTION file required CRAN to install hundreds of other packages to check caret. They were not pleased.

The Basic Release Process

1. create a few dynamic man pages
2. use `R CMD check --as-cran` to ensure passing CRAN tests and unit tests
3. update all packages (and R)
4. run regression tests and evaluate results
5. send to CRAN
6. repeat
7. repeat
8. install passed caret version
9. generate HTML documentation and sync github io branch
10. profit!

Required "Optimizations"

For example, there is one check that produces a large number of false positive warnings. For example:

```
> bwplot.diff.resamples <- function (x, data, metric = x$metric, ...)  {
+     ## some code
+     plotData <- subset(plotData, Metric %in% metric)
+     ## more code
+ }
```

will trigger a wanring that "`bwplot.diff.resamples: no visible binding for global variable 'Metric'`".

The "solution" is to have a file that is sourced first in the package (e.g. `aaa.R`) with the line

```
> Metric <- NULL
```

Severity of Problems

It's hard to tell which warnings should be ignored and which should not. There is also the issue of inconsistencies related to who is "on duty" when you submit your package.

For example, I recently updated the desirability package and received this warning:

Package Dependencies

This problem was somewhat alleviated at the end of 2013 when *custom methods* were introduced into the package.

Although this functionality had already existed in the package for some time, it was refactored to be more user freindly.

In the process, much of the modeling code was moved out of caret's R files and into R objects, eliminating the formal dependencies.

Right now, the *total* number of dependencies is much smaller (2 `Depends`, 7 `Imports`, and 25 `Suggests`).

This still affects testing though (described later). Also:

```
1 package is needed for this model and is not installed.  (gbm).
Would you like to try to install it now?
```

Regression Testing

Prior to CRAN release (or whenever required), a comprehensive set of regression tests are conducted.

All modeling packages are updated to their current CRAN versions.

For each model accessed by `train`, `rfe`, and/or `sbf`, a set of test cases are computed with the production version of caret and the devle version.

First, test cases are evaluated to make sure that nothing has been broken by updated versions of the consistuent packages.

Diffs of the model results are computed to assess any differences in caret versions.

This process takes approximately 3hrs to complete using `make -j 12` on a Mac Pro.

Regression Testing

```
$ R CMD BATCH make_model_Rd.R
$ cd ~/tmp/2015_04_19_09__6.0-41/
$ make -j 7 -i
 2015-04-19 09:13:44: Starting ada
 2015-04-19 09:13:44: Starting AdaBag
 2015-04-19 09:13:44: Starting AdaBoost.M1
 2015-04-19 09:13:44: Starting ANFIS
                    :
make: [FH.GBML.RData] Error 1 (ignored)
                    :
 2015-04-19 12:03:52: Finished WM
 2015-04-19 12:04:48: Finished xyf
```

Documentation

caret originally contained four package vignettes with in–depth descriptions of functionality with examples.

Although this functionality had already existed in the package for some time, it was refactored to be more user freindly.

However, this added time to `R CMD check` and was a general pain for CRAN.

Efforts to make the vingettes more computationally efficient (e.g. reducing the number of examples, resamples, etc.) diminished the effectiveness of the documentation.

Documentation

The documentation was moved out of the package and to the github IO page.

These pages are built using knitr whenever a new version is sent to CRAN. Some advantages are:

- longer and more relavant examples are availible
- update scheduile is under my control
- dynamic documentation (e.g. D3 network graphs, JS tables)
- better formatting

It currently takes about 4hr to create these (using parallel processing when possible).