lda denotes the linear discriminant analysis, *rpart* a classification tree, *bagging* bagging with 50 bootstrap samples, *double-bagging* bagging with 50 bootstrap samples, combined with LDA, *inclass-bagging* indirect classification using bagging and *inclass-lm* indirect classification using linear modeling.

Note that an estimator of the variance is available for the ordinary bootstrap estimator (`estimator="boot"`) only, see Efron and Tibshirani (1997).

Summary

ipred tries to implement a unified interface to some recent developments in classification and error rate estimation. It is by no means finished nor perfect and we very much appreciate comments, suggestions and criticism. Currently, the major drawback is speed. Calling `rpart` 50 times for each bootstrap sample is relatively inefficient but the design of interfaces was our main focus instead of optimization. Beside the examples shown, `bagging` can be used to compute bagged regression trees and `errorest` computes estimators of the mean squared error for regression models.

Bibliography

L. Breiman. Bagging predictors. *Machine Learning*, 24(2): 123–140, 1996a. 33

Leo Breiman. Out-of-bag estimation. Technical report, Statistics Department, University of California Berkeley, Berkeley CA 94708, 1996b. 33, 34

B. Efron and R. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. 33, 35, 36

D.J. Hand, H.G. Li, and N.M. Adams. Supervised classification with structured class definitions. *Computational Statistics & Data Analysis*, 36:209–225, 2001. 33, 34

T. Hothorn and B. Lausen. Double-bagging: Combining classifiers by bootstrap aggregation. submitted, 2002. preprint available under http://www.mathpreprints.com/. 34

T. Hothorn, I. Pal, O. Gefeller, B. Lausen, G. Michelson, and D. Paulus. Automated classification of optic nerve head topography images for glaucoma screening. In *Studies in Classification, Data Analysis, and Knowledge Organization (to appear)*. Proceedings of the 25th Annual Conference of the German Classification Society, 2002. 33

A. Peters, T. Hothorn, and B. Lausen. Glaucoma diagnosis by indirect classifiers. In *Studies in Classification, Data Analysis, and Knowledge Organization (to appear)*. Proceedings of the 8th Conference of the International Federation of Classification Societies, 2002. 33, 35

Friedrich-Alexander-Universität Erlangen-Nürnberg, Institut für Medizininformatik, Biometrie und Epidemiologie, Waldstraße 6, D-91054 Erlangen
Andrea.Peters@imbe.imed.uni-erlangen.de
Torsten.Hothorn@rzmail.uni-erlangen.de
Berthold.Lausen@rzmail.uni-erlangen.de

Support from Deutsche Forschungsgemeinschaft SFB 539-C1/A4 is gratefully acknowledged.

Changes in R

by the R Core Team

User-visible changes

- XDR support is now guaranteed to be available, so the default save format will always be XDR binary files, and it is safe to distribute data in that format. (We are unaware of any platform that did not support XDR in recent versions of R.)

  `gzfile()` is guaranteed to be available, so the preferred method to distribute sizeable data objects is now via `save(compress = TRUE)`.

- `pie()` replaces `piechart()` and defaults to using pastel colours.

- `formatC()` has new arguments (see below) and `formatC(*, d = <dig>)` is no longer valid and must be written as `formatC(*, digits = <dig>)`.

- Missingness of character strings is treated much more consistently, and the character string `"NA"` can be used as a non-missing value.

- `summary.factor()` now uses a stable sort, so the output will change where there are ties in the frequencies.

New features

- Changes in handling missing character strings:

  – `"NA"` is no longer automatically coerced to a missing value for a character string. Use `as.character(NA)` where a missing value is required, and test via `is.na(x)`, not `x`

== "NA". String "NA" is still converted to missing by scan() and read.table() unless 'na.strings' is changed from the default.

- A missing character string is now printed as 'NA' (no quotes) amongst quoted character strings, and '<NA>' if amongst unquoted character strings.
- axis() and text.default() omit missing values of their 'labels' argument (rather than plotting "NA").
- Missing character strings are treated as missing much more consistently, e.g., in logical comparisons and in sorts. identical() now differentiates "NA" from the missing string.

- Changes in package **methods**:
  - New function validSlotNames().
  - Classes can explicitly have a "data part", formally represented as a .Data slot in the class definition, but implemented consistently with informal structures. While the implementation is different, the user-level behavior largely follows the discussion in *Programming with Data*.
  - A "next method" facility has been provided, via the function callNextMethod(). This calls the method that would have been selected if the currently active method didn't exist. See ?callNextMethod. This is an extension to the API.
  - Classes can have initialize methods, which will be called when the function new() is used to create an object from the class. See ?initialize. This is an extension to the API.
  - The logic of setGeneric() has been clarified, simplifying nonstandard generic functions and default methods.

- Changes in package **tcltk**:
  - Now works with the GNOME user interface.
  - Several new functions allow access to C level Tcl objects. These are implemented using a new 'tclObj' class, and this is now the class of the return value from .Tcl() and tkcmd().

- Changes in package **ts**:
  - More emphasis on handling time series with missing values where possible, for example in acf() and in the ARIMA-fitting functions.
  - New function arima() which will replace arima0() in due course. Meanwhile, arima0() has been enhanced in several ways. Missing values are accepted. Parameter values can be initialized and can held fixed during fitting. There is a new argument 'method' giving the option to use conditional-sum-of-squares estimation.
  - New function arima.sim().
  - New datasets AirPassengers, Nile, UKgas and WWWusage, and a expanded version of UKDriverDeaths (as a multiple time series Seatbelts).
  - New generic function tsdiag() and methods for arima and arima0, to produce diagnostic plots. Supersedes arima0.diag().
  - New functions ARMAacf() and ARMAtoMA() to compute theoretical quantities for an ARMA process.
  - New function acf2AR() to compute the AR process with a given autocorrelation function.
  - New function StructTS() to fit structural time series, and new generic function tsSmooth() for fixed-interval state-space smoothing of such models.
  - New function monthplot() (contributed by Duncan Murdoch).
  - New functions decompose() and HoltWinters() (contributed by David Meyer) for classical seasonal decomposition and exponentially-weighted forecasting.

- An extensible approach to safe prediction for models with e.g. poly(), bs() or ns() terms, using the new generic function makepredictcall(). Used by most model-fitting functions including lm() and glm(). See ?poly, ?cars and ?ns for examples.

- acosh(), asinh(), atanh() are guaranteed to be available.

- axis() now omits labels which are NA (but still draws the tick mark.

- Connections to bzip2-ed files via bzfile().

- chol() allows pivoting via new argument 'pivot'.

- cmdscale() now takes rownames from a dist object 'd' as well as from a matrix; it has new arguments 'add' (as S) and 'x.ret'.

- `crossprod()` handles the case of real matrices with y = x separately (by accepting y = NULL). This gives a small performance gain (suggestion of Jonathan Rougier).
- `deriv()` and `deriv3()` can now handle expressions involving pnorm and dnorm (with a single argument), as in S-PLUS.
- New function `expm1()` both in R and in C API, for accurate $\exp(x) - 1$; precision improvement in `pexp()` and `pweibull()` in some cases. (PR#1334-5)
- New function `findInterval()` using new C entry point findInterval, see below.
- `formatDL()` now also works if both items and descriptions are given in a suitable list or matrix.
- `gzfile()` is guaranteed to be available, and hence the 'compress' option to `save()` and `save.image()`.
- `hist()` now has a method for date-time objects.
- `library()` now checks the dependence on R version (if any) and warns if the package was built under a later version of R.
- `library(help = PKG)` now also returns the information about the package PKG.
- Added function `logb()`, same as `log()` but for S-PLUS compatibility (where log now has only one argument).
- New na.action function `na.pass()` passes through NAs unaltered.
- `piechart()` has been renamed to `pie()`, as piechart is a Trellis function for arrays of pie charts. The default fill colours are now a set of pastel shades, rather than `par("bg")`.
- `plclust()` in package **mva**, for more S-PLUS compatibility.
- `poly()` now works with more than one vector or a matrix as input, and has a predict method for objects created from a single vector.
- `polyroot()` now handles coefficient vectors with terminal zeroes (as in S).
- New `prettyNum()` function used in `formatC()` and `format.default()` which have new optional arguments 'big.mark', 'big.interval', 'small.mark', 'small.interval', and 'decimal.mark'.
- `print.coefmat()` has a new argument 'eps.Pvalue' for determining when small P-values should be printed as '< {...}'.
- The `recover()` function has been moved to the **base** package. This is an interactive debugging function, usually a good choice for `options(error=)`. See `?recover`.
- `rep()` has a new argument 'each' for S-PLUS compatibility. The internal call is made available as `rep.int()`, again for help in porting code.
- New functions `rowSums()`, `colSums()`, `rowMeans()` and `colMeans()`: versions of `apply()` optimized for these cases.
- `rug()` now has a '`...`' argument allowing its location to be specified.
- `scan()` can have `NULL` elements in 'what', useful to save space when columns need to be discarded.
- New option 'by = "DSTday"' for `seq.POSIXt()`.
- Changes to sorting:
  - `sort()`, `sort.list()` and `order()` have a new argument 'decreasing' to allow the order to be reversed whilst still preserving ties.
  - `sort()` has an option to use quicksort in some cases (currently numeric vectors and increasing order).
  - The default Shell sort is Sedgewick's variant, around 20% faster, and pre-screening for NAs speeds cases without any NAs several-fold.
  - `sort.list()` (and order with just one vector) is several times faster for numeric, integer and logical vectors, and faster for character vectors.
- New assignment forms of `split()`; new function `unsplit()`.
- New `sprintf()` function for general C like formatting, from Jonathan Rougier.
- Argument 'split' of both `summary.aov` and `summary.aovlist` is now implemented.
- `summary.princomp()` now has a separate print method, and 'digits' is now an argument to the print method and not to `summary.princomp` itself.
- An extended version of the `trace()` function is available, compatible with the function in S-PLUS. Calls to R functions can be inserted on entry, on exit, and before any subexpressions. Calls to `browser()` and `recover()` are useful. See `?trace`.

- New function `TukeyHSD()` for multiple comparisons in the results of `aov()`. (Formerly function `Tukey` in package **Devore5** by Douglas Bates.)
- New read-only connections to files in zip files via `unz()`.
- `warning()` has new argument 'call.', like `stop()`'s.
- `zip.file.extract()` is no longer provisional and has an "internal" method available on all platforms.
- Methods for `[`, `[<-` and `as.data.frame()` for class `"POSIXlt"`.
- Much improved printing of matrices and arrays of type `"list"`.
- The "Knuth-TAOCP" option for random number generation has been given an option of using the 2002 revision. See ?RNG for the details: the R usage already protected against the reported 'weakness'.
- min/max of `integer(0)` (or `NULL`) is now `Inf`/`-Inf`, not an extreme integer.

Deprecated & defunct

- `.Alias()`, `reshapeLong()`, `reshapeWide()` are defunct.
- `arima0.diag()` (package **ts**) is deprecated: use `tsdiag()` instead.
- `piechart()` is deprecated; renamed to `pie()`.

Documentation changes

- *Writing R Extensions* now has an example of calling R's random numbers from FORTRAN via C.
- R itself and all R manuals now have ISBN numbers, please use them when citing R or one of the manuals.

Installation changes

- The configure script used when building R from source under Unix is now generated using Autoconf 2.50 or later, which has the following 'visible' consequences:
  - By default, configure no longer uses a cache file. Use the command line option '–config-cache' (or '-C') to enable caching.
  - Key configuration variables such as `CC` are now *precious*, implying that the variables
    - no longer need to be exported to the environment and can and should be set as command line arguments;
    - are kept in the cache even if not specified on the command line, and checked for consistency between two configure runs (provided that caching is used, see above);
    - are kept during automatic reconfiguration as if having been passed as command line arguments, even if no cache is used.

    See the variable output section of 'configure –help' for a list of all these variables.
- Configure variable `FC` is deprecated, and options '–with-g77', '–with-f77' and '–with-f2c' are defunct. Use configure variable `F77` to specify the FORTRAN 77 compiler, and `F2C` to specify the FORTRAN-to-C compiler and/or that it should be used even if a FORTRAN 77 compiler is available.
- Non-standard directories containing libraries are specified using configure variable `LDFLAGS` (not `LIBS`).

Utilities

- `Sweave()`, `Stangle()` and friends in package **tools**. Sweave allows mixing LaTeX documentation and R code in a single source file: the R code can be replaced by its output (text, figures) to allow automatic report generation. Sweave files found in package subdir 'inst/doc' are automatically tested by `R CMD check` and converted to PDF by `R CMD build`, see the section on package vignettes in *Writing R Extensions*.
- Rdconv can convert to the S4 '.sgml' format.
- 'R::Utils.pm' masks some platform dependencies in Perl code by providing global variables like `R_OSTYPE` or wrapper functions like `R_runR()`.
- If a directory 'inst/doc' is present in the sources of a package, the HTML index of the installed package has a link to the respective subdirectory.
- `R CMD check` is more stringent: it now also fails on malformed 'Depends' and 'Maintainer' fields in 'DESCRIPTION' files, and on unbalanced braces in Rd files. It now also provides pointers to documentation for problems it reports.

- `R CMD check`, `build` and `INSTALL` produce outline-type output.
- QA functions in package **tools** now return the results of their computations as objects with suitable `print()` methods. By default, output is only produced if a problem was found.
- New utility `R CMD config` to get the values of basic R configure variables, or the header and library flags necessary for linking against R.
- `Rdindex` and 'maketitle.pl' require Perl 5.005, as 'Text::Wrap::fill' was only introduced at 5.004_05.

C-level facilities

- All the double-precision BLAS routines are now available, and package writers are encouraged not to include their own (so enhanced ones will be used if requested at configuration).
- `findInterval(xt[],n,x,...)` gives the index (or interval number) of `x` in the sorted sequence `xt[]`. There's an `F77_SUB(interv)(.)` to be called from FORTRAN; this used to be part of `predict.smooth.spline`'s underlying FORTRAN code.
- Substitutes for (v)snprintf will be used if the OS does not supply one, so tests for `HAVE_(V)SNPRINTF` are no longer needed.
- The DUP and NAOK arguments in a `.C()` call are not passed on to the native routine being invoked. Any code that relied on the old behaviour will need to be modified.
- `log1p` is only provided in 'Rmath.h' if it is not provided by the platform, in which case its name is not remapped, but a back-compatibility entry point `Rf_log1p` is provided. Applications using libRmath may need to be re-compiled.
- The methods used by `integrate()` and `optim()` have entry points in 'R_ext/Applic.h' and have a more general interface documented in *Writing R Extensions*.
- The bessel_? entry points are now suitable to be called repeatedly from code loaded by `.C()`. (They did not free memory until `.C()` returned in earlier versions of R.)
- Server sockets on non-Windows platforms now set the SO_REUSEADDR socket option. This allows a server to create simultanous connections to several clients.
- New quicksort sorting (for numeric no-NA data), accessible from C as `R_qsort()` etc and from FORTRAN as `qsort4()` and `qsort3()`.
- 'Rinternals.h' no longer includes 'fcntl.h', as this is not an ISO C header and cannot be guaranteed to exist.
- FORTRAN subroutines are more correctly declared as 'extern void' in 'R_exts/Applic.h' and 'R_exts/Linpack.h'.

Bug fixes

- The calculation of which axes to label on a `persp()` plot was incorrect in some cases.
- Insufficient information was being recorded in the display list for the `identify()` function. In particular, the 'plot' argument was ignored when replaying the display list. (PR#1157)
- The vertical alignment of mathematical annotations was wrong. When a vertical adjustment was not given, it was bottom-adjusting i.e,. it was treating `adj=0` as `adj=c(0, 0)`. It now treats `adj=0` as `adj=c(0, 0.5)` as for "normal" text. (PR#1302)
- the man page ('doc/R.1') wasn't updated with the proper R version.
- `smooth.spline()` had a 'df = 5' default which was never used and hence extraneous and misleading.
- `read.fwf()` was interpreting comment chars in its call to scan: replaced by a call to `readLines()`. (PR#1297/8)
- The default comment char in `scan()` has been changed to `""` for consistency with earlier code (as in the previous item).
- `bxp(*, notch.frac = f)` now draws the median line correctly.
- Current versions of `gs` were rotating the output of `bitmap(type = "pdfwrite")` and when converting the output of `postscript()` to PDF; this has been circumvented by suppressing the '%%Orientation' comment for non-standard paper sizes.
- `plot.ts(x, log = "y")` works again when `x` has 0s, also for matrix `x`.
- `add1()`, `drop1()`, `step()` work again on glm objects with formulae with rhs's containing '.'. (Broken by a 'bug fix' (in reality an API change) in 1.2.1.)
- `optim(method="BFGS")` was not reporting reaching 'maxit' iterations in the convergence component of the return value.

- `aov()` and `model.tables()` were failing on multistrata models with excessively long Error formula. (PR#1315)
- Transparent backgrounds on `png()` devices on Unix-alikes had been broken during the driver changes just prior to 1.4.0. (They worked correctly on Windows.)
- `demo(is.things)` didn't work properly when the methods package was attached.
- `match()`, `unique()` and `duplicated()` were not declaring all `NaN`s to be equal, yet not always distinguishing `NA` and `NaN`. This was very rare except for data imported as binary numbers.
- The error handler `recover()` protects itself against errors in dump.frames and uses a new utility, limitedLabels, to generate names for the dump that don't inadvertently blow the limit on symbol length. (TODO: either fix dump.frames accordingly or remove the limit– say by truncating very long symbols?)
- `se.contrasts()` works more reliably with multistratum models, and its help page has an example.
- `summary.lm()` was not returning r.squared nor adj.r.squared for intercept-only models, but `summary.lm.null()` was returning r.squared but not adj.r.squared. Now both are always returned. Neither returned f.statistic, and that is now documented.
- Subsetting of matrices of mode `"list"` (or other non-atomic modes) was not implemented and gave incorrect results without warning. (PR#1329). Under some circumstances subsetting of a character matrix inserted `NA` in the wrong place.
- `abs()` was not being treated as member of the Math group generic function, so e.g. its method for data frames was not being used.
- `set.seed(seed, "default")` was not using the 'seed' value (only for 'kind = "default"').
- `logLik.lm()` now uses 'df = p + 1' again ('+ sigma'!).
- `logLik.glm()` was incorrect for families with estimated dispersion.
- Added `strptime()` workaround for those platforms (such as Solaris) that returned missing components as 0. Missing days are now detected, but missing years will still be interpreted as 1900 on such platforms.
- Inheritance in formal classes (the **methods** package) works breadth-first as intuition would expect.
- The `new()` function in package **methods** works better (maybe even correctly?) for the various combinations of super-classes and prototypes that can be supplied as unnamed arguments.
- Internal code allowed one more connection to be allocated than the table size, leading to segfaults. (PR#1333)
- If a user asks to open a connection when it is created and it cannot be opened, the connection is destroyed before returning from the creation call. (related to PR#1333)
- `Sys.putenv()` was not using permanent storage. (PR#1371)
- `La.svd()` was not coercing integer matrices. (PR#1363)
- `deriv(3)` now reports correctly the function it cannot find the derivatives table.
- The GNOME user interface was overenthusiastic about setting locale information. Now only `LC_CTYPE`, `LC_COLLATE` and `LC_TIME` are determined by the user's environment variables (PR#1321).
- In X11, `locator()` would sound the bell even if `xset b off` had been set.
- `merge()` could be confused by inconsistent use of `as.character()` giving leading spaces.
- `[pqr]binom()` no longer silently round the 'size' argument, but return `NaN` (as `dbinom()` does). (PR#1377)
- Fixed socket writing code to block until all data is written. Fixed socket reading code to properly handle long reads and reads with part of the data in the connection buffer.
- Allow sockets to be opened in binary mode with both 'open="ab"' and 'open="a+b"'.
- `levels<-.factor()` was using incorrectly list values longer than the number of levels (PR#1394), and incorrectly documented that a character value could not be longer than the existing levels.
- The `pdf()` device was running out of objects before the documented 500 page limit. Now there is no limit.
- `legend()` did not deal correctly with 'angle' arguments. (PR#1404)
- `sum()` tried to give an integer result for integer arguments, but (PR#1408)

  - this was not documented
  - it sometimes warned on overflow, sometimes not
  - it was order-dependent for a mixture of integer and numeric args.

- `mean()` gave (numeric) `NA` if integer overflow occurred in `sum()`, but now always works internally with numeric (or complex) numbers.
- `sort.list()` and `order()` were treating NA_STRING as "NA".
- `sort.list(na.last = NA)` was not implemented.
- `seq.default()` was returning only one element for a relative range of less than about 1e-8, which was excessively conservative. (PR#1416)
- `tsp(x) <- NULL` now also works after attaching the **methods** package.
- `persp(shade=)` was not working correctly with the default col=NULL if this was transparent. (PR#1419)
- `min(complex(0))` and `max(complex(0))` were returning random values.
- `range()` gave `c(1, 1)`.
- `range(numeric(0))` is now `c(Inf, -Inf)`, as it was documented to be.
- `print.ts()` was occasionally making rounding errors in the labels for multiple calendar time series.
- `Rdconv` was not handling nested `\describe{}` constructs in conversion to HTML (PR#1257) and not fixing up mal-formed `\item` fields in `\describe{}` in conversion to text (PR#1330).
- `filled.contour()` was not checking consistency of x, y, z. (PR#1432)
- `persp.default()` no longer crashes with non-character labels. (PR#1431)
- `fft()` gave incorrect answers for input sizes 392, 588, 968, 980, ... (PR#1429)
- `det(method = "qr")` gave incorrect results for numerically singular matrices. (PR#1244)
- `barplot()` now allows the user to control 'xpd'. (PR#1088, 1398)
- `library()` (with no arguments) no longer fails on empty 'TITLE' files.
- `glm()` was failing if both `offset()` and start were specified. (PR#1421)
- `glm()` might have gotten confused if both step-shortening and pivoting had occurred (PR#1331). Step-halving to avoid the boundary of feasible values was not working.
- Internal representation of logical values was not being treated consistently. (Related to PR#1439)
- The `c()` function sometimes inserted garbage in the name vector for some types of objects, e.g. `names(c(ls, a=1))`.
- Fixed bug in '`$`' that could cause mutations on assignment (PR#1450).
- Some X servers displayed random bytes in the window title of graphics windows (PR#1451)
- The X11 data editor would segfault if closed with window manager controls (PR#1453)
- Interrupt of `Sys.sleep()` on UNIX no longer causes subsequent `Sys.sleep()` calls to segfault due to infinite recusion.
- Eliminated a race condition that could cause segfaults when a SIGINT was received while handling an earlier SIGINT.
- `rect(lty = "blank")` was incorrectly drawing with a dashed line.
- `type.convert()` was not reporting incorrectly formatted complex inputs. (PR#1477)
- `readChar()` was not resetting vmax, so causing memory build-up. (PR#1483)