Journal of Statistical Software
April 2010, Volume 34, Issue 7. http://www.jstatsoft.org/

rpartOrdinal: An R Package for Deriving a Classification Tree for Predicting an Ordinal Response

Kellie J. Archer
Virginia Commonwealth University

Abstract

This paper describes an R package, **rpartOrdinal**, that implements alternative splitting functions for fitting a classification tree when interest lies in predicting an ordinal response. This includes the generalized Gini impurity function, which was introduced as a method for predicting an ordinal response by including costs of misclassification into the impurity function, as well as an alternative ordinal impurity function due to Piccarreta (2008) that does not require the assignment of misclassification costs. The ordered twoing splitting method, which is not defined as a decrease in node impurity, is also included in the package. Since, in the ordinal response setting, misclassifying observations to adjacent categories is a less egregious error than misclassifying observations to distant categories, this package also includes a function for estimating an ordinal measure of association, the gamma statistic.

Keywords: machine learning, classification trees, recursive partitioning, ordinal response, R.

1. Introduction

For many high-throughput genomic studies, the phenotype to be predicted is ordinal. Some examples of ordinal responses include the more recently advocated method for evaluating response to treatment in target tumor lesions, known as the response evaluation criteria in solid tumors (RECIST) method, with ordinal outcomes defined as complete response > partial response > stable disease > progressive disease. Moreover, most histopathological measures are ordinal, such as scoring methods for liver biopsy specimens from patients with chronic hepatitis, including the Knodell hepatic activity index, the Ishak score, and the METAVIR score. Statistical methods such as adjacent category, proportional odds, and continuation ratio models (Agresti 2002) are traditionally used when modeling an ordinal response, though

they fail for high-throughput genomic datasets when the number of covariates, p, exceeds the number of observations, n.

An alternative class prediction method, classification trees (CTs), is capable of predicting a response when the $n << p$ (Breiman *et al.* 1984). Suppose n independent observations to be classified are characterized by a p-dimensional vector of predictors $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and each observation $\boldsymbol{x}_i$ falls into one of J classes. Let ω denote the class with $\omega = \omega_1$ representing observations in class 1, $\omega = \omega_2$ representing class 2,..., and $\omega = \omega_J$ representing class J. When deriving a CT, all observations start together in the root node, t. Then, for predictors $1, 2, \ldots, p$, the optimal split is determined, where optimality is defined as that split resulting in the largest decrease in node impurity.

For node t, the optimal split divides the observations to the left and right descendent nodes, t_L and t_R, respectively, and the proportion of cases in each of the J classes within these nodes are called the node proportions, that is, $p(\omega_j|t)$ for $j = 1, \ldots, J$ such that $p(\omega_1|t) + p(\omega_2|t) + \ldots + p(\omega_J|t) = 1$. For nominal response classification, the within-node impurity measure most commonly used is the Gini criterion (Breiman *et al.* 1984), defined as

$$i(t) = \sum_k \sum_{k \neq l} p(\omega_k|t) p(\omega_l|t). \tag{1}$$

This is the default impurity function in the R programming environment (R Development Core Team 2009) **rpart** package (Therneau and Atkinson 1997) for predicting a nominal class response. However, use of this impurity function does not take advantage of the additional information present when the response is ordinal. To that end, the generalized Gini impurity function (Breiman *et al.* 1984),

$$i_{GG}(t) = \sum_k \sum_{k \neq l} C(\omega_k|\omega_l) p(\omega_k|t) p(\omega_l|t), \tag{2}$$

which factors in $C(\omega_k|\omega_l)$, the cost of misclassifying a class l observation as belonging to class k, has been proposed for ordinal response prediction.

Another proposed ordinal impurity function for deriving an ordinal response classification tree based on a measure of nominal-ordinal association (Piccarreta 2001) that does not require the assignment of costs of misclassification is

$$i_{OS}(t) = \sum_{j=1}^{J} F(\omega_j|t) \left(1 - F(\omega_j|t)\right) \tag{3}$$

where $F(\omega_j|t) = \sum_{k=1}^{j} p(\omega_k|t)$ (Piccarreta 2008).

A splitting method for ordinal response prediction that is not impurity-based is the ordered twoing method (Breiman *et al.* 1984). Though this method was described in the seminal book by Breiman *et al.* (1984) and has been implemented in the CART Software by Salford Systems (Steinberg and Colla 1997; Steinberg and Golovnya 2006), it has not yet been implemented in R. Ordered twoing proceeds by reformulating the ordinal response as a vector of dichotomous responses, where for each observation i, the j-th dichotomous response is taken to be

$$C_{ij} = \begin{cases} 1 & \text{if } \omega_i = 1, \ldots, j \\ 0 & \text{if } \omega_i = j+1, \ldots, J. \end{cases} \tag{4}$$

For node t and dichotomous response C_j, the split s that maximizes

$$\phi(s,t,C_j) = 2p_L p_R (p(C_j|t_L) - p(C_j|t_R))^2 \tag{5}$$

over the p covariates is taken to be the best split for that dichotomous response C_j. Subsequently, the split s associated with the dichotomous response

$$j^* = \arg\max_j \phi(s,t,C_j) \tag{6}$$

is then selected for splitting node t. In this paper, we describe the **rpartOrdinal** R package, which implements ordered twoing, the generalized Gini, and the ordinal impurity splitting methods. These splitting methods should be considered for use when deriving an ordinal response classification tree.

For nominal response prediction, misclassification rates are often examined as a means for assessing the performance of the classifier. For ordinal response prediction problems, it may be of more interest to estimate the gamma statistic as an ordinal measure of association between the observed and predicted responses as a means for gauging the success of ordinal classification. Briefly, the association between two ordinal variables X and Y can be estimated by the gamma statistic (Agresti 2002), where given the cross-tabulation matrix T of X and Y having r rows and c columns, the number of concordant pairs for cells $(1,1)$ to $(r-1,c-1)$ is given by

$$C_{kl} = T_{kl} \times \sum_{j=l+1}^{c} \sum_{i=k+1}^{r} T_{ij}. \tag{7}$$

Similarly, the number of discordant pairs for cells $(1,2)$ to $(r-1,c)$ is given by

$$D_{kl} = T_{kl} \times \sum_{j=1}^{l-1} \sum_{i=k+1}^{r} T_{ij}. \tag{8}$$

We then let $C = \sum_{j=1}^{c-1} \sum_{i=1}^{r-1} C_{ij}$ and $D = \sum_{j=2}^{c} \sum_{i=1}^{r-1} D_{ij}$ such that the gamma statistic of ordinal association is defined as

$$\hat{\gamma} = \frac{C-D}{C+D}. \tag{9}$$

An R package, **rpartOrdinal**, that implements the described ordinal splitting methods as well as a function for estimating the gamma statistic as an ordinal measure of association is available for download from the Comprehensive R Archive Network at http://CRAN.R-project.org/package=rpartOrdinal.

2. Illustrative datasets

2.1. Low birthweight dataset

The `lowbwt` dataset was downloaded from ftp://ftp.wiley.com/public/sci_tech_med/logistic/ and includes birthweight and associated risk factors measured on 189 women as described by (Hosmer and Lemeshow 2000). For illustrative purposes, an ordinal response variable (`Category`) will be derived from birthweight as defined in Table 1 and added to the `lowbwt` dataset.

1	`bwt` $>$ 3500
2	3000 $<$ `bwt` $\leq$ 3500
3	2500 $<$ `bwt` $\leq$ 3000
4	`bwt` $\leq$ 2500

Table 1: Ordinal response levels for low birthweight (`Category`).

Variable	Description
`low`	Dichotomous outcome: Low birthweight (<2,500 grams) or not
`age`	Age of mother, years
`lwt`	Mother's weight at last menstrual period, pounds
`race`	Race of mother (white, black, other)
`smoke`	Mother's smoking status (No, Yes)
`ptl`	Number of previous premature labours
`ht`	Mother's history of hypertension (No, Yes)
`ui`	Presence of uterine irritability (No, Yes)
`ftv`	Number of physician visits during the first trimester
`bwt`	Birth weight in grams

Table 2: Description of covariates included in the low birthweight dataset.

In addition to this ordinal response, the dataset includes variables listed in Table 2.

2.2. Gene expression in B-cell acute lymphocytic leukemia

As an example using a high-througput genomic dataset, the acute lymphoblastic leukemia `ALL` dataset was downloaded from the **Bioconductor** experiment repository `http://www.bioconductor.org/packages/release/data/experiment/html/ALL.html`, which includes gene expression microarray data for 128 patients, 95 with B-cell and 33 with T-cell leukemia (Chiaretti *et al.* 2004, 2005). Patients with B-cell acute lymphocytic leukemia (B-ALL) are staged according to whether or not the leukemic cells express different antigens (e.g., CD19, HLA-DR, CD10) or immunoglobins (e.g., surface immunoglobin or cytoplasmic immunoglobins). The stages are ordered, such that B1 represents early pre-B ALL (do not express CD10); B2 represents disease more advanced than B1 as CD10 is expressed, but not yet meeting criteria stages B3-B4; B3 represents common ALL, where the CD10 antigen is expressed but still lacking IgM; and B4 represents pre-B ALL, where CD10 and IgM are expressed. B-ALL stage is clinically important as it is one of several factors used to plan treatment. Among the 95 B-ALL patients in the publicly available dataset, 90 were staged (19 B1, 36 B2, 23 B3, and 12 B4 patients). Note that this dataset requires more memory than is typically available on a standard Windows PC. The B-ALL dataset was therefore analyzed using a MacBook Pro laptop (Mac OS X 10.5.7) having 8GB RAM.

3. Implementation

The **rpartOrdinal** package was written in the R programming environment (R Development

Core Team 2009) and depends on the **rpart** package (Therneau and Atkinson 1997). Currently, **rpart** includes methods for deriving regression, classification, and survival trees. Due to the `method =` option in **rpart**, users can define their own splitting methods for use in conjunction with the `rpart` function. A user defined method passed to the `method =` option must be a list consisting of three functions named `eval`, `split`, and `init`. Since previous research comparing the ordinal splitting methods to traditional methods for single trees and for bootstrap aggregating classification trees has demonstrated that when the response to be predicted is ordinal, an ordinal splitting method is usually preferred (Piccarreta 2008; Archer and Mas 2009), we have implemented three ordinal splitting methods, namely, ordered twoing, ordinal impurity, and generalized Gini for use in conjunction with **rpart**.

3.1. Ordered twoing

The ordered twoing splitting criteria in Equation 5 has been implemented as a callable method in **rpart**. Here we derive an ordinal classification tree for predicting the ordinal response in the low birthweight dataset, `Category`, using ordered twoing by additionally specifying `method = twoing`. Such a CT may be useful for exploring factors related to poor neonatal outcomes.

```
R> library("rpartOrdinal")
R> data("lowbwt")
R> lowbwt$Category <- factor(ifelse(lowbwt$bwt <= 2500, 3,
+    ifelse(lowbwt$bwt <= 3000, 2,
+    ifelse(lowbwt$bwt <= 3500, 1, 0))), ordered = TRUE)
R> otwoing.rpart <- rpart(Category ~ age + lwt + race + smoke + ptl +
+    ht + ui + ftv, data = lowbwt, method = twoing)
```

The fitted CT can be graphically displayed using the `plot()` and `text()` functions. In plots of the tree topology created using these functions, observations meeting the criterion displayed at a given node proceed to the left descendent node while observations not meeting the criterion displayed at a given node proceed to the right descendent node.

```
R> plot(otwoing.rpart)
R> text(otwoing.rpart, pretty = TRUE)
```

The additional `pretty = TRUE` argument to the `text` function does not abbreviate factors as alphanumeric characters if they appear in the tree. Alternatively, the `post()` function can be used for producing postscript files containing the tree topology which more extensively labels the splits and predicted class for each node.

```
R> post(otwoing.rpart, filename = "TwoingLowbwt.ps", use.n = FALSE,
+    title = "", horizontal = FALSE)
```

3.2. Ordinal impurity function

As with the twoing function, the ordinal impurity function in Equation 3 has been implemented as a callable method in **rpart**. That is, within the `rpart` function, the user should specify `method = ordinal` to fit an ordinal response classification tree using Equation 3.

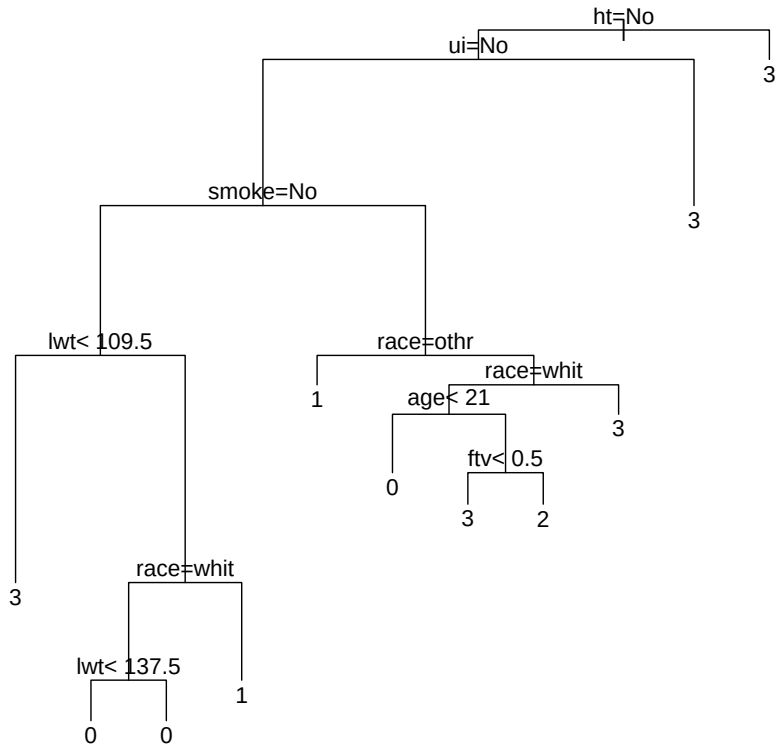

Figure 1: CT for low birthweight dataset using ordered twoing.

```
R> ordinal.rpart <- rpart(Category ~ age + lwt + race + smoke + ptl +
+    ht + ui + ftv, data = lowbwt, method = ordinal)
R> plot(ordinal.rpart)
R> text(ordinal.rpart, pretty = TRUE)
```

The ordered twoing and ordinal tree topologies are very similar with exception that node 38 is split by `lwt` in the ordinal tree whereas this same node is split by `age` in the ordered twoing tree, with the descendent node 77 splitting variable also differing.

3.3. Generalized Gini impurity

The generalized Gini impurity function in Equation 2 has been implemented in this package by allowing the user to specify a `loss.matrix` parameter in the optional `parms` argument within the `rpart` call. The `loss.matrix` parameter accepts either `"linear"` or `"quadratic"` for using either linear or quadratic loss, respectively. The specific syntax for the low birthweight example using the linear loss follows.

```
R> linear.loss.rpart <- rpart(Category ~ age + lwt + race + smoke + ptl +
+    ht + ui + ftv, data = lowbwt, method = "class",
+    parms = list(loss = loss.matrix(method = "linear", lowbwt$Category)))
R> plot(linear.loss.rpart)
R> text(linear.loss.rpart, pretty = TRUE)
```

The specific syntax for the low birthweight example using the quadratic loss function is

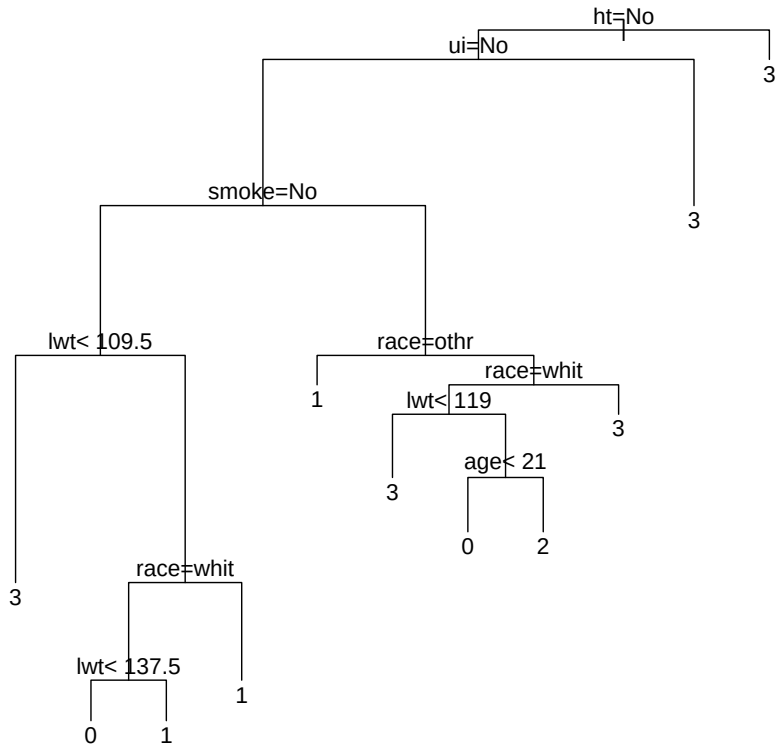

Figure 2: CT for low birthweight dataset using the ordinal impurity function.

```
R> quad.loss.rpart <- rpart(Category ~ age + lwt + race + smoke + ptl +
+    ht + ui + ftv, data = lowbwt, method = "class",
+    parms = list(loss = loss.matrix(method = "quad", lowbwt$Category)))
R> plot(quad.loss.rpart)
R> text(quad.loss.rpart, pretty = TRUE)
```

Both CTs derived using the generalized Gini criteria split the root node using `lwt>=109.5` and split node 2 using `ui = 0`. However, other splits differed between the linear and quadratic loss functions.

3.4. Gamma statistic

The `ordinal.gamma` function estimates the gamma statistic, which is a measure of the strength of the association of the cross-tabulation of two ordinal variables. The following example replicates Table 2.8 in Agresti (2002).

```
R> library("rpartOrdinal")
R> job.satis <- factor(c(1, rep(2, 3), rep(3, 10), rep(4, 6), rep(1, 2),
+    rep(2, 3), rep(3, 10), rep(4, 7), 1, rep(2, 6), rep(3, 14), rep(4, 12),
+    2, rep(3, 9), rep(4, 11)), ordered = TRUE,
+    labels = c("Very Dissatisfied", "Little Dissatisfied",
+    "Moderately Satisfied", "Satisfied"))
R> income <- factor(c(rep(1, 20), rep(2, 22), rep(3, 33), rep(4, 21)),
```

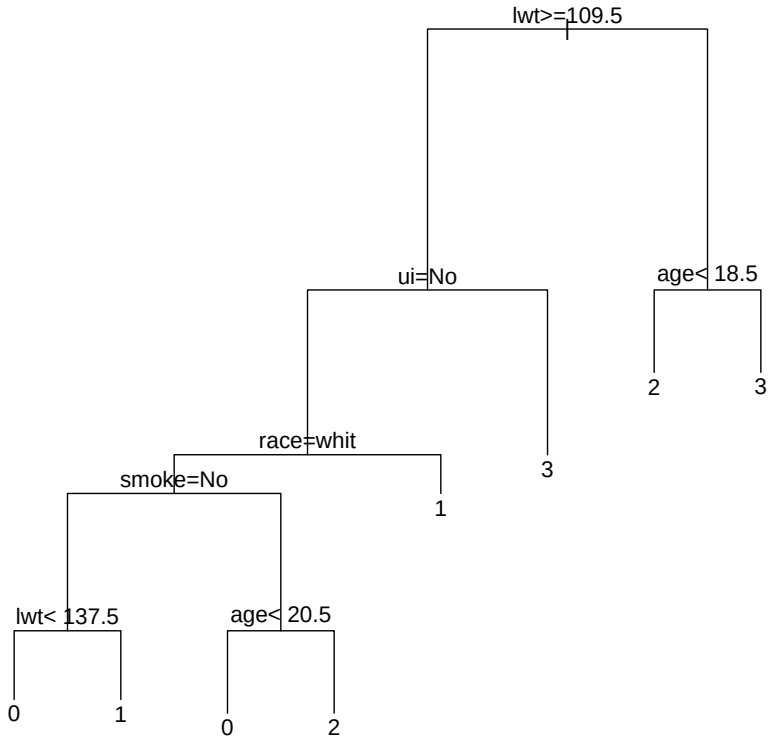

Figure 3: CT for lowbwt using generalized Gini with linear cost of misclassification.

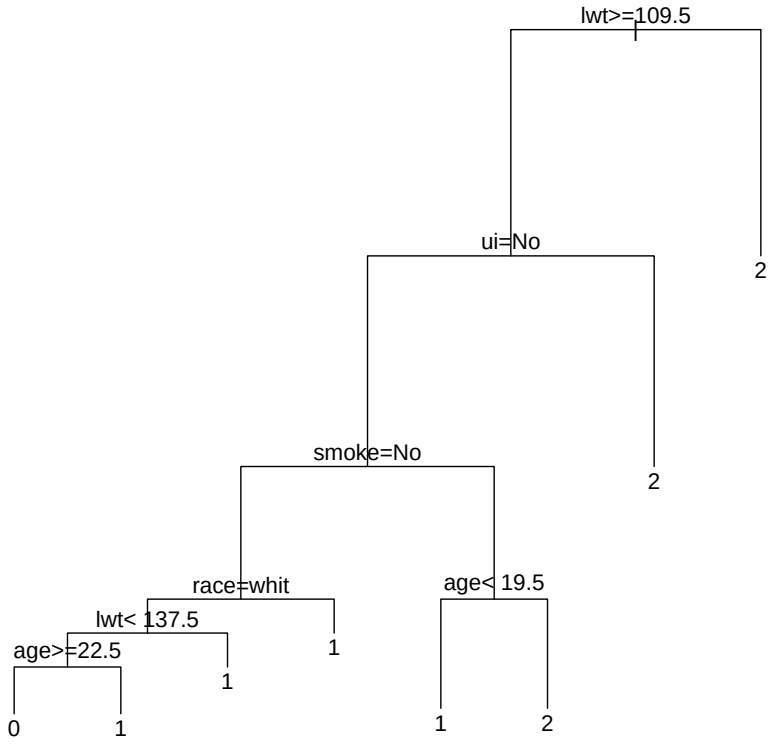

Figure 4: CT for lowbwt using generalized Gini with quadratic cost of misclassification.

```
+    ordered = TRUE, labels = c("<15,000", "15,000-25,000", "25,000-40,000",
+    ">40,000"))
R> table(job.satis, income)

                     income
job.satis             <15,000 15,000-25,000 25,000-40,000 >40,000
  Very Dissatisfied         1             2             1       0
  Little Dissatisfied       3             3             6       1
  Moderately Satisfied     10            10            14       9
  Satisfied                 6             7            12      11

R> ordinal.gamma(job.satis, income)

[1] 0.2211009
```

Returning to the ordinal classification trees for the low birthweight dataset, it may be of interest to estimate the gamma statistic as an ordinal measure of association between the observed and predicted ordinal responses. However, estimating the gamma statistic as an ordinal measure of association for the training data will not provide useful information regarding how well the predictor may generalize when presented with new data. Therefore, cross-validation methods may be used. The following code was used to perform five-fold cross-validation where the observations included in the V-th fold are stored in the V-th component of `groups`. Letting $\mathcal{L}$ represent the full dataset, each method (ordinal, ordered twoing, generalized Gini with linear loss, and generalized Gini with quadratic loss) was fit using the observations in $\mathcal{L} \setminus \mathcal{L}_v$ then the predicted class was obtained for the observations in $\mathcal{L}_v$.

```
R> V <- 5
R> n <- length(lowbwt$Category)
R> leave.out <- trunc(n/V)
R> o <- sample(1:n)
R> groups <- vector("list", V)
R> for(j in 1:(V - 1)) {
+    jj <- (1 + (j - 1) * leave.out)
+    groups[[j]] <- (o[jj:(jj + leave.out - 1)])
+  }
R> groups[[V]] <- o[(1 + (V - 1) * leave.out):n]
R> linear.fit <- rep(NA, n)
R> quad.fit <- rep(NA, n)
R> ordinal.fit <- rep(NA, n)
R> twoing.fit <- rep(NA, n)
R> for(j in 1:V) {
+    ordinal.rpart <- rpart(Category ~ age + lwt + race + smoke +
+      ptl + ht + ui + ftv, data = lowbwt, subset = -groups[[j]],
+      method = ordinal)
+    ordinal.fit[groups[[j]]] <- predict(ordinal.rpart,
+      newdata = lowbwt[groups[[j]],])
+    twoing.rpart <- rpart(Category ~ age + lwt + race + smoke +
```

Ordinal impurity	0.446
Ordered twoing	0.436
Generalized Gini with linear cost of misclassification	0.345
Generalized Gini with quadratic cost of misclassification	0.402

Table 3: Five-fold cross-validation estimate of the gamma ordinal association measure between the observed and predicted ordinal response for the low birthweight dataset.

```
+      ptl + ht + ui + ftv, data = lowbwt, subset = -groups[[j]],
+      method = twoing)
+    twoing.fit[groups[[j]]] <- predict(twoing.rpart,
+      newdata = lowbwt[groups[[j]],])
+    linear.rpart <- rpart(Category ~ age + lwt + race + smoke +
+      ptl + ht + ui + ftv, data = lowbwt, subset= -groups[[j]],
+      parms = list(loss = loss.matrix(method = "linear", lowbwt$Category)))
+    phat <- predict(linear.rpart, newdata=lowbwt[groups[[j]],])
+    linear.fit[groups[[j]]] <- apply(phat, 1, which.max)
+    quadratic.rpart <- rpart(Category ~ age + lwt + race + smoke +
+      ptl + ht + ui + ftv, data = lowbwt, subset = -groups[[j]],
+      parms = list(loss = loss.matrix(method = "quad", lowbwt$Category)))
+    phat <- predict(quadratic.rpart, newdata=lowbwt[groups[[j]],])
+    quad.fit[groups[[j]]] <- apply(phat, 1, which.max)
+  }
R> ordinal.gamma(lowbwt$Category, twoing.fit)
R> ordinal.gamma(lowbwt$Category, ordinal.fit)
R> ordinal.gamma(lowbwt$Category, linear.fit)
R> ordinal.gamma(lowbwt$Category, quad.fit)
```

For this random partition of the `lowbwt` dataset, the ordinal and ordered twoing methods had the similar performance and both performed better than the generalized Gini with either quadratic or linear loss (Table 3). If the sample size is large, a split sample approach could be used wherein the `rpart` function would be applied to a `train` dataset and the `predict` function applied using `newdata=test`. Alternatively, one can easily construct a bootstrap procedure and estimate error using out-of-bag observations as in Archer and Mas (2009).

3.5. Gene expression in B-ALL

Here we demonstrate application of the ordinal classification methods for predicting B-ALL stage.

```
R> library("rpartOrdinal")
R> library("ALL")
R> data("ALL")
```

The class object `ALL` is an ExpressionSet, developed by the **Bioconductor** project (Gentleman *et al.* 2004) as a container for high-throughput genomic datasets. This object includes both a $g \times n$ matrix gene expression data, where g represents the number of probesets (i.e., genes)

interrogated by the high-throughput assay and n represents the number of samples processed, and an $n \times p$ data frame of phenotypic data, where again n represents the number of samples and p represents the number of phenotypic variables. The gene expression matrix can be extracted from the `ALL` object using the `exprs()` extractor function while the phenotypic data can be accessed using the `pData()` extractor function. As described in section 2.2, the `ALL` object includes gene expression and phenotypic data for 128 patients, 95 with B-cell and 33 with T-cell leukemia. In this example, we will restrict attention to only those patients with B-cell leukemia who were also staged as either B1, B2, B3, or B4. The `pData(ALL)` object includes a vector `BT` which stores the type (B or T) and stage (1, 2, 3, or 4) of disease and can be used for subsetting the `ALL` object. The following line of code was used to restrict the dataset to patients staged as B1, B2, B3, or B4. Further information on the phenotypic variables stored in the `ALL` object can be obtained by issuing `?ALL`.

```
R> BALL <- ALL[, is.element(pData(ALL)$BT, c("B1", "B2", "B3", "B4"))]
```

Next we construct an ordered factor `stage` to represent B-ALL stage as our ordinal outcome.

```
R> stage <- factor(pData(BALL)$BT, levels = c("B1", "B2", "B3", "B4"),
+    ordered = TRUE)
```

Ordinal CTs predicting disease stage may be useful for exploring genetic mechanisms that lead to B-ALL progression. Prior to fitting a CT, a data frame consisting of the ordinal outcome `stage` and the transposed $g \times n$ gene expression matrix must be constructed.

```
R> Bcell <- data.frame(t(exprs(BALL)), stage)
```

Once the data frame has been constructed, ordinal classification trees may be fit using syntax similar to the `lowbwt` example. The following syntax was used to fit CTs using ordered twoing, ordinal, generalized Gini with linear loss, and generalized Gini with quadratic loss, respectively.

```
R> otwoing.rpart <- rpart(stage ~ ., data = Bcell, method = twoing)
R> plot(otwoing.rpart)
R> text(otwoing.rpart)
R> ord.rpart <- rpart(stage ~ ., data = Bcell, method = ordinal)
R> plot(ord.rpart)
R> text(ord.rpart)
R> linear.loss.rpart <- rpart(stage ~ ., data = Bcell,
+    parms = list(loss = loss.matrix(method = "linear", Bcell$stage)))
R> plot(linear.loss.rpart)
R> text(linear.loss.rpart)
R> quad.loss.rpart <- rpart(stage ~ ., data = Bcell,
+    parms = list(loss = loss.matrix(method = "quad",Bcell$stage)))
R> plot(quad.loss.rpart)
R> text(quad.loss.rpart)
```

Interestingly, all four methods split the root node using the same variable and cutpoint. For ordinal classification, it may be of interest to use five-fold cross-validation to estimate the gamma statistic as an ordinal measure of association between the observed and predicted ordinal responses. The following code was used for performing five-fold cross-validation.

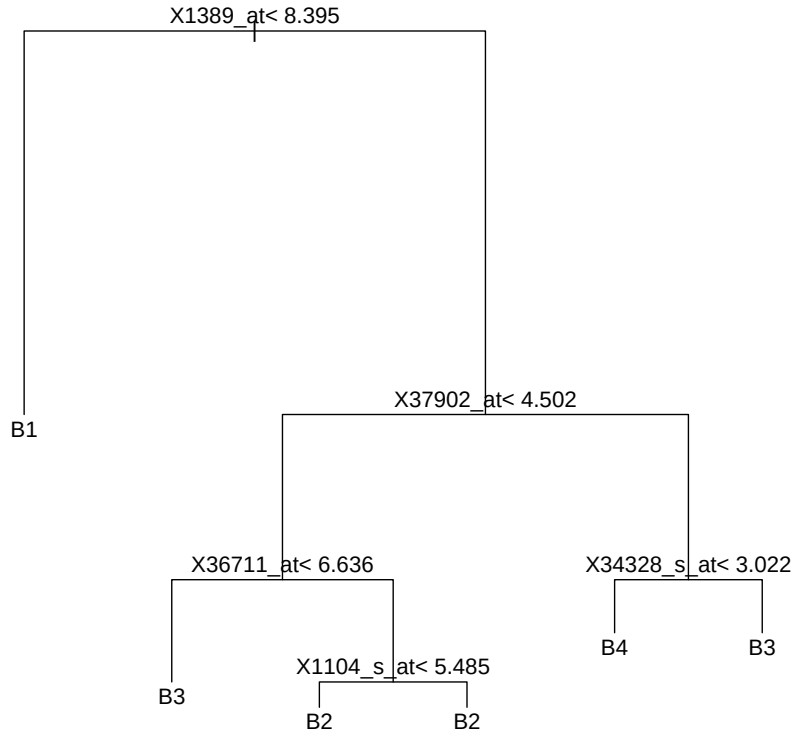

Figure 5: CT for B-ALL using ordered twoing.

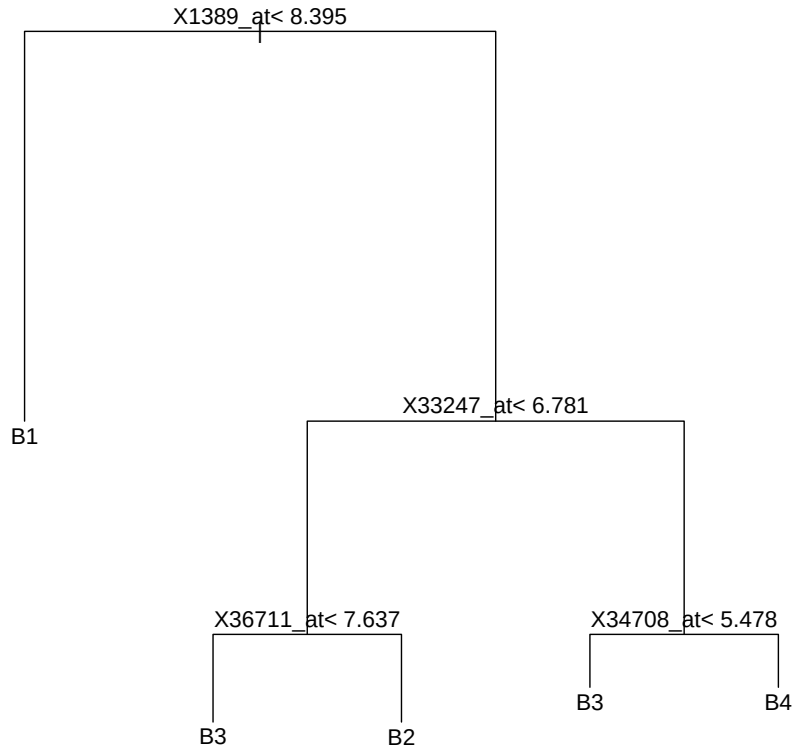

Figure 6: CT for B-ALL using the ordinal impurity function.

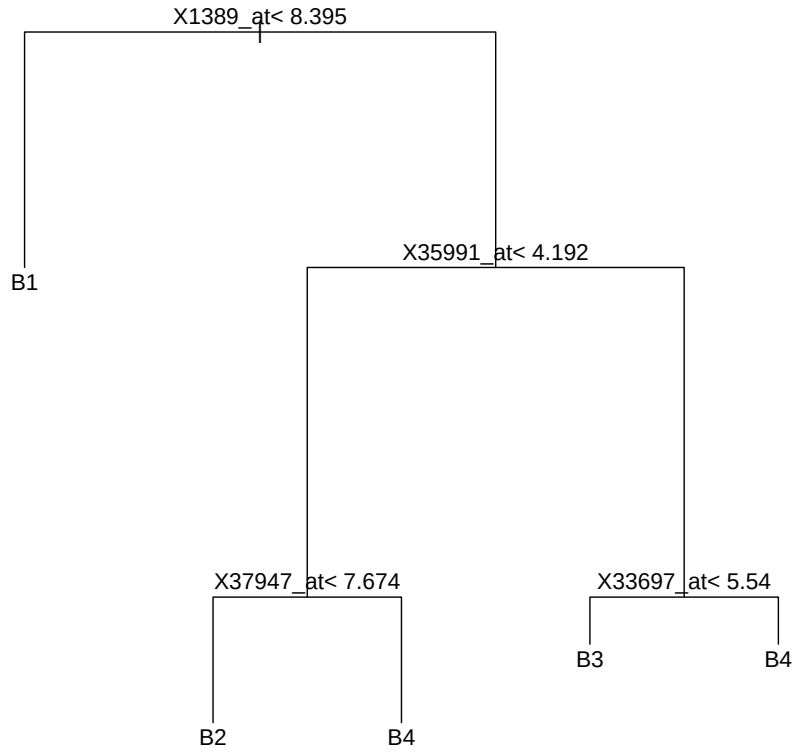

Figure 7: CT for B-ALL using generalized Gini with linear cost of misclassification.

```
R> V <- 5
R> n <- length(Bcell$stage)
R> leave.out <- trunc(n/V)
R> o <- sample(1:n)
R> groups <- vector("list", V)
R> for(j in 1:(V - 1)) {
+    jj <- (1 + (j - 1) * leave.out)
+    groups[[j]] <- (o[jj:(jj + leave.out - 1)])
+  }
R> groups[[V]] <- o[(1 + (V - 1) * leave.out):n]
R> linear.fit <- rep(NA, n)
R> quad.fit <- rep(NA, n)
R> ordinal.fit <- rep(NA, n)
R> twoing.fit <- rep(NA, n)
R> for(j in 1:V) {
+    train <- Bcell[-groups[[j]],]
+    ordinal.rpart <- rpart(stage ~ ., data = train, method = ordinal)
+    twoing.rpart <- rpart(stage ~ ., data = train, method = twoing)
+    linear.rpart <- rpart(stage ~ ., data = train,
+      parms = list(loss = loss.matrix(method = "linear", train$stage)))
+    quad.rpart <- rpart(stage ~ ., data = train,
+      parms = list(loss = loss.matrix(method = "quad", train$stage)))
```

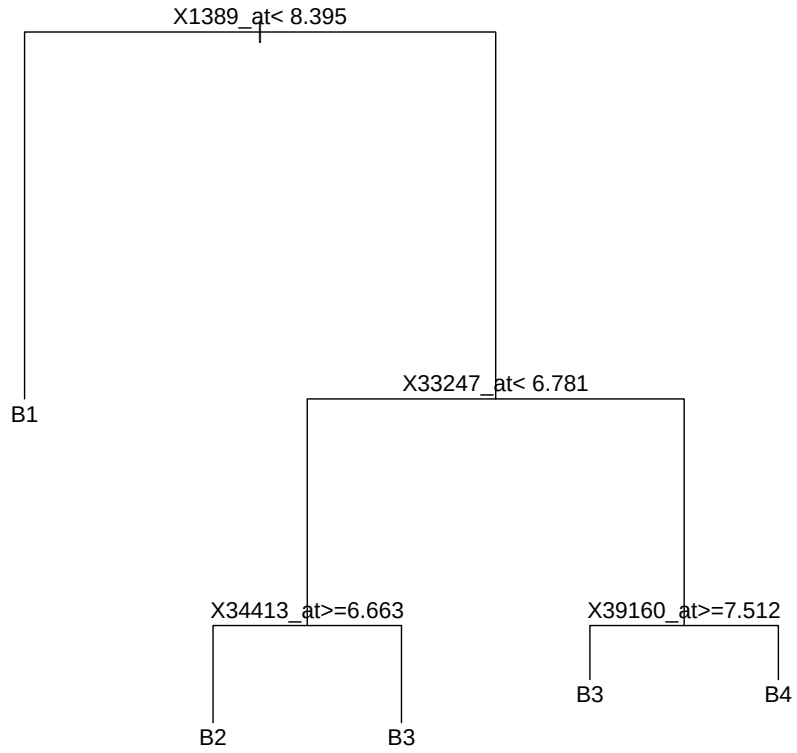

Figure 8: CT for B-ALL using generalized Gini with quadratic cost of misclassification.

```
+    rm(train)
+    test <- Bcell[groups[[j]],]
+    ordinal.fit[groups[[j]]] <- predict(ordinal.rpart, newdata = test)
+    twoing.fit[groups[[j]]] <- predict(twoing.rpart, newdata = test)
+    phat <- predict(linear.rpart, newdata = test)
+    linear.fit[groups[[j]]]<-apply(phat, 1, which.max)
+    rm(phat)
+    phat <- predict(quad.rpart, newdata = test)
+    quad.fit[groups[[j]]] <- apply(phat, 1, which.max)
+    rm(ordinal.rpart, twoing.rpart, linear.rpart, quad.rpart, phat, test)
+  }
R> ordinal.gamma(Bcell$stage, ordinal.fit)
R> ordinal.gamma(Bcell$stage, twoing.fit)
R> ordinal.gamma(Bcell$stage, linear.fit)
R> ordinal.gamma(Bcell$stage, quad.fit)
```

The gamma ordinal association measure for each of the four splitting methods for the B-ALL gene expression dataset are listed in Table 4.

Ordinal impurity	0.762
Ordered twoing	0.562
Generalized Gini with linear cost of misclassification	0.736
Generalized Gini with quadratic cost of misclassification	0.687

Table 4: Five-fold cross-validation estimate of the gamma ordinal association measure between the observed and predicted ordinal response for for the B-ALL dataset.

4. Summary

Herein we have described the **rpartOrdinal** package which works in conjuction with the **rpart** package in the R programming environment. The package provides methods for fitting a CT when the response is ordinal. We note that another R package, **party** (Hothorn *et al.* 2009), can also be used to derive an ordinal conditional inference tree, where the variable selected for splitting a given node is determined using an inferential test (Hothorn *et al.* 2006). These methods may prove useful when the dataset to be analyzed includes an ordinal response and the number of covariates exceeds the sample size. In such situations, traditional ordinal response methods such as proportional odds models cannot be fit. Secondary data analyses are a natural and desirable by-product from publicly available databases such as Gene Expression Omnibus. In the high-throughput genomic setting, most attention has been focused on classification algorithms for dichotomous responses. We believe analysts will find the **rpartOrdinal** useful particularly when modeling ordinal responses for high-dimensional datasets.

Acknowledgments

This research was supported by the National Institute of Library Medicine R03LM009347.

References

Agresti AA (2002). *Categorical Data Analysis*. 2nd edition. John Wiley & Sons, Hoboken, NJ.

Archer KJ, Mas VR (2009). "Ordinal Response Prediction Using Bootstrap Aggregation, with Application to a High-throughput Methylation Dataset." *Statistics in Medicine*, **28**, 3597–3610.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA.

Chiaretti S, Li X, Gentleman R, Vitale A, Vignetti M, Mandelli F, Ritz J, Foá R (2004). "Gene Expression Profile of Adult T-cell Acute Lymphocytic Leukemia Identifies Distinct Subsets of Patients with Different Response to Therapy and Survival." *Blood*, **103**, 2771–2778.

Chiaretti S, Li X, Gentleman R, Vitale A, Wang K, Mandelli F, Foá R, Ritz J (2005). "Gene Expression Profles of B-lineage Adult Lymphocytic Leukemia Reveal Genetic Patterns that Identify Lineage Derivation and Distinct Mechanisms of Transformation." *Clinical Cancer Research*, **20**, 7209–7219.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "**Bioconductor**: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**, R80. URL http://genomebiology.com/2004/5/10/R80.

Hosmer DW, Lemeshow S (2000). *Applied Logistic Regression*. 2nd edition. John Wiley & Sons, New York.

Hothorn T, Hornik K, Strobl C, Zeileis A (2009). ***party**: A Laboratory for Recursive Partytioning*. R package version 0.9-999, URL http://CRAN.R-project.org/package=party.

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics*, **15**(3), 651–674.

Piccarreta R (2001). "A New Measure of Nominal-Ordinal Association." *Journal of Applied Statistics*, **28**, 107–120.

Piccarreta R (2008). "Classification Trees for Ordinal Variables." *Computational Statistics*, **23**, 407–427.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Steinberg D, Colla P (1997). *CART-Classification and Regression Trees*. Salford Systems, San Diego, CA.

Steinberg D, Golovnya M (2006). *CART 6.0 User's Manual*. Salford Systems, San Diego, CA.

Therneau TM, Atkinson EJ (1997). "An Introduction to Recursive Partitioning Using the **rpart** Routine." *Technical Report 61*, Section of Biostatistics, Mayo Clinic, Rochester. URL http://www.mayo.edu/hsr/techrpt/61.pdf.

Affiliation:

Kellie J. Archer
Department of Biostatistics

Virginia Commonwealth University
730 East Broad Street
Richmond, Virginia, 23298-0032, United States of America
E-mail: kjarcher@vcu.edu
URL: http://www.people.vcu.edu/~kjarcher/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 34, Issue 7 *Submitted:* 2009-08-07
April 2010 *Accepted:* 2010-03-18