

- Rodríguez, G. (2008). Multilevel generalized linear models. In *Handbook of multilevel analysis* (pp. 335–376). New York: Springer.
- Rokach, L., & Maimon, O. (2005). Decision trees. In *Data mining and knowledge discovery handbook* (pp. 165–192). US: Springer. doi:10.1007/0-387-25465-X\_9
- Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, 87(418), 407–418. doi:10.2307/2290271
- Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning*, 86(2), 169–207. doi:10.1007/s10994-011-5258-3
- Snijders, T., & Bosker, J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. CA: Sage Publications.
- Steinberg, D., & Colla, P. (2009). CART: classification and regression trees. In *The top ten algorithms in data mining* (pp. 179–201). New York, NY: CRC Press.
- Steyerberg, E. W., Harrell, F. E., Jr., Borsboom, G. J., Eijkemans, M. J. C., Vergouwe, Y., & Habbema, J. D. F. (2001). Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. *Journal of Clinical Epidemiology*, 54(8), 774–781. doi:10.1016/S0895-4356(01)00341-9
- Timofeev, R. (2004). *Classification and regression trees (CART) theory and applications* (Doctoral dissertation). Humboldt University, Berlin.
- Verbyla, D. L. (1987). Classification trees: A new discrimination tool. *Canadian Journal of Forest Research*, 17(9), 1150–1152. doi:10.1139/x87-177
- Wong, G. Y., & Mason, W. M. (1985). The hierarchical logistic regression model for multilevel analysis. *Journal of the American Statistical Association*, 80(391), 513–524. doi:10.1080/01621459.1985.10478148
- Yu, B. (2012). Variable selection and adjustment in relation to propensity scores and prognostic scores: From single-level to multilevel data (Doctoral dissertation). University of Toronto, Toronto, Canada.
- Zhang, H. (1998). Classification trees for multiple binary responses. *Journal of the American Statistical Association*, 93(441), 180–193. doi:10.1080/01621459.1998.10474100
- Zhang, H., & Ye, Y. (2008). A tree-based method for modeling a multivariate ordinal response. *Statistics and Its Interface*, 1(1), 169–178. doi:10.4310/SII.2008.v1.n1.a14

## Appendix A

### The proposed M.CART algorithm in R

```
M.CART <- function (formula, data, random="SID",
  rslope="+X1 + X2", lgmodel="int", initialRandomEffects =
  rep(0, TotalObs), ErrorTolerance = 0.00001, MaxIterations =
  10000, verbose = FALSE, cpmin = 0.01) {
```

```
  TotalObs <- dim(data)[1]
  originaldata <- data
  Predictors <- paste(attr(terms(formula), "term.labels"),
    collapse = "+")
  TargetName <- formula[[2]]
  if (length(TargetName) > 1) TargetName <- TargetName[3]
  if (verbose) print(paste("Target variable: ", TargetName))
  Target <- data[, toString(TargetName)]
```

```
  newdata <- data
```

```
  originaldata$random <- rep(0, TotalObs)
  for (i in 1:length(summary(originaldata$TID))) {
    originaldata$random[originaldata$TID == i] <-
    mean(originaldata[originaldata$TID==i, toString(Target-
    Name)])-
    mean(originaldata[, toString(TargetName)])
  }
```

```
  AdjustedTarget <- data[, toString(TargetName)]-
  originaldata$random
  ContinueCondition <- TRUE
  iterations <- 0
  oldlik <- 0
```

```
  while (ContinueCondition) {
    iterations <- iterations + 1
    newdata[, "AdjustedTarget"] <- AdjustedTarget
    tree <- rpart(formula(paste(c("AdjustedTarget",
      Predictors), collapse = "~")), data =
    newdata, method = "anova", control = rpart.control
    (cp = cpmin, xval = 10))
    if (verbose)
      print(tree)
```

```
    newdata[, "nodeInd"] <- tree$where
    if (min(tree$where) == max(tree$where)) {
      glmerfit2 <- glmer(formula(paste(c(toString(TargetName),
        paste("1+(1|", random, ")", sep="")), collapse = "~")), data =
        newdata, family = binomial, nAGQ = 0, na.action =
        na.exclude)
    } else {
      if (lgmodel=="int") {
        glmerfit2 <- glmer(formula(paste(c(toString(TargetName),
          paste("as.factor(nodeInd)+(1|", random, ")", sep="")),
          collapse = "~")), data = newdata,
          family = binomial, nAGQ = 0, na.action = na.exclude)
      } else {
        glmerfit2 <- glmer(formula(paste(c(toString
          (TargetName), paste("as.factor(nodeInd)", paste
          ("(1", rslope, "|", random, ")", sep=""), sep="+")),
          collapse = "~")), data = newdata, family = binomial,
          nAGQ = 0, na.action = na.exclude)
      }
    }
  }
```

```
  newlik <- logLik(glmerfit2)
  ContinueCondition <- (abs(newlik - oldlik) >
    ErrorTolerance & iterations < MaxIterations)
  oldlik <- newlik
  AdjustedTarget2 <- (
    as.matrix(getME(glmerfit2, name="X"))
    %*% (as.matrix(getME(glmerfit2, name="beta")))
  )
  AdjustedTarget <- exp(AdjustedTarget2)/
    (1 + exp(AdjustedTarget2))
  AdjustedTarget[is.na(AdjustedTarget[, 1]), 1] <- (data[,
    toString(TargetName)]-
    originaldata$random)[is.na(AdjustedTarget[, 1])]
}
```

```
if (lgmodel!="int") {
  Between <- cbind(VarCorr(glmerfit2)[[1]][1:2],
    VarCorr(glmerfit2)[[1]][3:4])
}
```

```

} else {
  Between<-VarCorr(glmerfit2)[[1]][1]
}
preditFinal<-predict(glmerfit2,type="response")

result <- list(data = data,IterationsUsed = iterations,
  Random = random, Rslope = rslope,
  ErrorTolerance = ErrorTolerance, prune_cv = prune_cv,
  predMtree = preditFinal,
  Tree = tree, Treewhere = tree$where, EffectModel =
  glmerfit2, RandomEffects = ranef(glmerfit2),
  BetweenMatrix = as.matrix(Between)*sigma(glmerfit2)^2,
  logLik = newlik,
  AIC = AIC(glmerfit2), BIC = BIC(glmerfit2), deviance =
  deviance(glmerfit2),
  df = as.numeric(summary(glmerfit2)$AICtab[5]),
  Formula = formula,Totalinter = iterations)

```

```

class(result) <- "M.CART"
return(result)
}

```

## Appendix B

### *The R code for the empirical demonstration*

```

M.CART(Y ~ ReadScore + MathScore + ScienceScore + EnjoySch
+ CloseT + CloseS + GradeImp + Home + FlGood +
ParentEdu + ParentTlk + ParentHlp + Public + PctHsp +
PctBlk + SchFund, data = train, random="schoolID", lgmo-
del="slope", rslope="+ ReadScore + MathScore +
ScienceScore + CloseT + ParentEdu * SchFund",
cpmin =0.001)

```