An example of variable selection in `GLMcat`

Lorena León* Jean Peyhardi† Catherine Trottier‡

This vignette we will explore different models, evaluate their performance using Akaike Information Criterion (AIC), and select the best model based on AIC values.

For the models using the *Student CDF* and the *Logistic CDF*, we start by defining the full model formula as **`choice ~ hinc[air] + psize[air] + gc + ttme`**. We then fit the full model using **`discrete_cm`** and calculate its AIC. Additionally, we fit an intercept-only model and compare its AIC with that of the full model to determine if the full model provides a better fit.

Next, we perform backward elimination to simplify the full model by iteratively removing one predictor at a time. We fit reduced models for each predictor elimination and store the best model and its AIC value. After completing the backward elimination, we obtain the final model that offers the best fit based on AIC.

```
# Load required packages
library(GLMcat)

# Load the data
data(TravelChoice)

# Define the full model formula
full_model_formula <- choice ~ hinc[air] + psize[air] + gc + ttme
```

Student CDF

Model Selection Algorithm:

Step 1: Fit Model 0 - Intercept Model (AIC0) This model assumes a constant intercept. It serves as the baseline for model comparison.

```
mod0 <- discrete_cm(
  formula = choice ~ 1,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod0 <- AIC(mod0)
```

Step 2: Fit Model 1.1 - Intercept + Hinc (AIC1.1) This model includes the variable 'hinc' as a predictor in addition to the intercept.

*Université de Montpellier, ylorenaleonv@gmail.com
†Université de Montpellier, jean.peyhardi@umontpellier.fr
‡Université de Montpellier, catherine.trottier@umontpellier.fr

```
mod11 <- discrete_cm(
  formula = choice ~ hinc[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod11 <- AIC(mod11)
```

Step 3: Fit Model 1.2 - Intercept + Psize (AIC1.2) This model includes the variable 'psize' as a predictor in addition to the intercept.

```
mod12 <- discrete_cm(
  formula = choice ~ psize[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod12 <- AIC(mod12)
```

Step 4: Fit Model 1.3 - Intercept + GC (AIC1.3) This model includes the variable 'gc' as a predictor in addition to the intercept.

```
mod13 <- discrete_cm(
  formula = choice ~ gc,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod13 <- AIC(mod13)
```

Step 5: Fit Model 1.4 - Intercept + Ttime (AIC1.4) This model includes the variable 'ttme' as a predictor in addition to the intercept.

```
mod14 <- discrete_cm(
  formula = choice ~ ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod14 <- AIC(mod14)
```

Step 6: Select the model with the minimum AIC value from the first set of models (AIC1)

```
aic_values_1 <- c(aicmod11, aicmod12, aicmod13, aicmod14)
AIC1 <- min(aic_values_1)
which_min_AIC1 <- which.min(aic_values_1)
which_min_AIC1
```

```
## [1] 4
```

Step 7: Check if AIC1 is smaller than AIC0 to decide whether to continue with Model 2

```
AIC0_is_smaller_than_AIC1 <- AIC1 < aicmod0
AIC0_is_smaller_than_AIC1
```

```
## [1] TRUE
```

Step 8: Fit Model 2.1 - Intercept + Ttime + Hinc (AIC2.1) This model includes 'ttme' and 'hinc' as additional predictors along with the intercept.

```
mod21 <- discrete_cm(
  formula = choice ~ hinc[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod21 <- AIC(mod21)
```

Step 9: Fit Model 2.2 - Intercept + Ttime + Psize (AIC2.2) This model includes 'ttme' and 'psize' as additional predictors along with the intercept.

```
mod22 <- discrete_cm(
  formula = choice ~ psize[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod22 <- AIC(mod22)
```

Step 10: Fit Model 2.3 - Intercept + Ttime + GC (AIC2.3) This model includes 'ttme' and 'gc' as additional predictors along with the intercept.

```
mod23 <- discrete_cm(
  formula = choice ~ gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmod23 <- AIC(mod23)
```

Step 11: Select the model with the minimum AIC value from the second set of models (AIC2)

```
aic_values_2 <- c(aicmod21, aicmod22, aicmod23)
AIC2 <- min(aic_values_2)
which_min_AIC2 <- which.min(aic_values_2)
which_min_AIC2
```

```
## [1] 2
```

Step 12: Check if AIC2 is greater than AIC1 to decide whether to stop the algorithm

```
AIC2_is_greater_than_AIC1 <- AIC2 > AIC1
AIC2_is_greater_than_AIC1
```

```
## [1] TRUE
```

We then stop here and select only Ttme as the nly explanatory variable.

Backward Algorithm: Starting from choice ~ hinc[air] + psize[air] + gc + ttme, eliminate one predictor at a time and compare AIC values Step 13: Fit Model with all predictors (hinc[air], psize[air], gc, ttme)

```
modc0 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc0 <- AIC(modc0)
```

Step 14-16: Fit Model without predictors hinc[air], psize[air], and gc

```
modc11 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + gc,
  case_id = "indv",
  alternatives = "mode",
```

```
  reference = "car",
  alternative_specific = c("gc"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc11 <- AIC(modc11)


modc12 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc12 <- AIC(modc12)


modc13 <- discrete_cm(
  formula = choice ~ hinc[air] + gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc13 <- AIC(modc13)
```

Step 17: Select the model with the minimum AIC value among the backward models (AICC1)

```
aic_values_backward <- c(aicmodc11, aicmodc12, aicmodc13)
AICC1 <- min(aic_values_backward)
which_min_AICC1 <- which.min(aic_values_backward)
which_min_AICC1
```

```
## [1] 2
```

```
AICC1_is_smaller_than_all_predictors <- AICC1 < aicmodc0
AICC1_is_smaller_than_all_predictors
```

```
## [1] TRUE
```

Step 18-20: Continue the backward algorithm Fit Model

```
modc21 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc21 <- AIC(modc21)

modc22 <- discrete_cm(
  formula = choice ~ hinc[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc22 <- AIC(modc22)

modc23 <- discrete_cm(
  formula = choice ~ psize[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc23 <- AIC(modc23)
```

Step 21: Select the model with the minimum AIC value among the backward models (AICC2)

```
aic_values_backward_2 <- c(aicmodc21, aicmodc22, aicmodc23)
AICC2 <- min(aic_values_backward_2)
which_min_AICC2 <- which.min(aic_values_backward_2)
which_min_AICC2
```

```
## [1] 3
```

```
AICC2_is_smaller_than_AICC1 <- AICC2 < AICC1
AICC2_is_smaller_than_AICC1
```

```
## [1] TRUE
```

Step 22-23: Continue the backward algorithm Fit Model

```
modc31 <- discrete_cm(
  formula = choice ~ psize[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc31 <- AIC(modc31)

modc32 <- discrete_cm(
  formula = choice ~ ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = list("student"),
  find_nu = TRUE
)
aicmodc32 <- AIC(modc32)
```

Step 24: Select the model with the minimum AIC value among the backward models (AICC3)

```
aic_values_backward_3 <- c(aicmodc31, aicmodc32)
AICC3 <- min(aic_values_backward_3)
which_min_AICC3 <- which.min(aic_values_backward_3)
which_min_AICC3
```

```
## [1] 2
```

```
AICC3_is_smaller_than_AICC2 <- AICC3 < AICC2
AICC3_is_smaller_than_AICC2
```

```
## [1] TRUE
```

Best performance for Model with ttme as the only predictor

Logistic CDF

Step 0: Model 0 - Intercept Model (AIC0)

```
mod0 <- discrete_cm(
  formula = choice ~ 1,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = "logistic"
)
mod0$cdf
```

```
## $cdf
## [1] "logistic"
##
## $freedom_degrees
## [1] 1
##
## $mu
## [1] 0
```

```
aicmod0 <- AIC(mod0)
```

Step 1: Model 1.1 - Intercept + Hinc (AIC1.1)

```
mod11 <- discrete_cm(
  formula = choice ~ hinc[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = "logistic"
)
mod11$cdf
```

```
## $cdf
## [1] "logistic"
##
## $freedom_degrees
## [1] 1
##
## $mu
## [1] 0
```

```
aicmod11 <- AIC(mod11)
```

Step 2: Model 1.2 - Intercept + Psize (AIC1.2)

```
mod12 <- discrete_cm(
  formula = choice ~ psize[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  data = TravelChoice,
  cdf = "logistic"
)
mod12$cdf
```

```
## $cdf
## [1] "logistic"
##
## $freedom_degrees
## [1] 1
##
## $mu
## [1] 0
```

```
aicmod12 <- AIC(mod12)
```

Step 3: Model 1.3 - Intercept + GC (AIC1.3)

```
mod13 <- discrete_cm(
  formula = choice ~ gc,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc"),
  data = TravelChoice,
  cdf = "logistic"
)
mod13$cdf
```

```
## $cdf
## [1] "logistic"
##
## $freedom_degrees
## [1] 1
##
## $mu
## [1] 0
```

```
aicmod13 <- AIC(mod13)
```

Step 4: Model 1.4 - Intercept + Ttime (AIC1.4)

```
mod14 <- discrete_cm(
  formula = choice ~ ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
mod14$cdf
```

```
## $cdf
## [1] "logistic"
##
## $freedom_degrees
## [1] 1
##
## $mu
## [1] 0
```

```
aicmod14 <- AIC(mod14)
```

Step 5: Select the model with the minimum AIC value from the first set of models (AIC1)

```
aic_values_1 <- c(aicmod11, aicmod12, aicmod13, aicmod14)
AIC1 <- min(aic_values_1)
which_min_AIC1 <- which.min(aic_values_1)
which_min_AIC1
```

```
## [1] 4
```

Step 6: Check if AIC1 is smaller than AIC0 to decide whether to continue with Model 2

```
AIC0_is_smaller_than_AIC1 <- AIC1 < aicmod0
AIC0_is_smaller_than_AIC1
```

```
## [1] TRUE
```

Step 7: Model 2.1 - Intercept + Ttime + Hinc (AIC2.1)

```
mod21 <- discrete_cm(
  formula = choice ~ hinc[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
```

```
aicmod21 <- AIC(mod21)
```

Step 8: Model 2.2 - Intercept + Ttime + Psize (AIC2.2)

```
mod22 <- discrete_cm(
  formula = choice ~ psize[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
```

```
aicmod22 <- AIC(mod22)
```

Step 9: Model 2.3 - Intercept + Ttime + GC (AIC2.3)

```
mod23 <- discrete_cm(
  formula = choice ~ gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
```

```
aicmod23 <- AIC(mod23)
```

Step 10: Select the model with the minimum AIC value from the second set of models (AIC2)

```
aic_values_2 <- c(aicmod21, aicmod22, aicmod23)
AIC2 <- min(aic_values_2)
which_min_AIC2 <- which.min(aic_values_2)
which_min_AIC2
```

```
## [1] 3
```

Step 11: Check if AIC2 is greater than AIC1 to decide whether to stop the algorithm

```
AIC2_is_greater_than_AIC1 <- AIC2 > AIC1
AIC2_is_greater_than_AIC1
```

```
## [1] FALSE
```

Step 12: Model 2.3.1 - Intercept + Ttime + GC + Hinc (AIC2.3.1)

```
mod231 <- discrete_cm(
  formula = choice ~ gc + ttme + hinc[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
aicmod231 <- AIC(mod231)
```

Step 13: Model 2.3.2 - Intercept + Ttime + GC + Psize (AIC2.3.2)

```
mod232 <- discrete_cm(
  formula = choice ~ gc + ttme + psize[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
aicmod232 <- AIC(mod232)
```

Step 14: Select the model with the minimum AIC value among the additional models (AIC3)

```
AIC3 <- min(aicmod231, aicmod232)
which_min_AIC3 <- which.min(c(aicmod231, aicmod232))
which_min_AIC3
```

```
## [1] 2
```

```
AIC3_is_smaller_than_AIC2 <- AIC3 < AIC2
AIC3_is_smaller_than_AIC2
```

```
## [1] TRUE
```

Step 15: Model 2.3.2.1 - Intercept + Ttime + GC + Psize + Hinc (AIC2.3.2.1)

```
mod2321 <- discrete_cm(
  formula = choice ~ gc + ttme + psize[air] + hinc[air],
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)
aicmod2321 <- AIC(mod2321)
```

Step 16: Check if the AIC of Model 2.3.2.1 is smaller than AIC3

```
aicmod2321_is_smaller_than_AIC3 <- aicmod2321 < AIC3
aicmod2321_is_smaller_than_AIC3
```

```
## [1] TRUE
```

We obtain then the full model with the logistic CDF

Step 17: Backwards algorithm, starting with choice ~ hinc[air] + psize[air] + gc + ttme

```
modc0 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)

aicmodc0 <- AIC(modc0)
aicmodc0
```

```
## [1] 385.8297
```

```
modc11 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + gc,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc"),
```

```
  data = TravelChoice,
  cdf = "logistic"
)

aicmodc11 <- AIC(modc11)

modc12 <- discrete_cm(
  formula = choice ~ hinc[air] + psize[air] + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("ttme"),
  data = TravelChoice,
  cdf = "logistic"
)

aicmodc12 <- AIC(modc12)

modc13 <- discrete_cm(
  formula = choice ~ hinc[air] + gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)

aicmodc13 <- AIC(modc13)

modc14 <- discrete_cm(
  formula = choice ~ psize[air] + gc + ttme,
  case_id = "indv",
  alternatives = "mode",
  reference = "car",
  alternative_specific = c("gc", "ttme"),
  data = TravelChoice,
  cdf = "logistic"
)

aicmodc14 <- AIC(modc14)
```

Step 18: Select the model with the minimum AIC value among the backward models (AICC1)

```
aic_values_backward <- c(aicmodc11, aicmodc12, aicmodc13, aicmodc14)
AICC1 <- min(aic_values_backward)
which_min_AICC1 <- which.min(aic_values_backward)
which_min_AICC1
```

```
## [1] 4
```

```
AICC1_is_smaller_than_aicmodc0 <- AICC1 < aicmodc0
AICC1_is_smaller_than_aicmodc0
```

```
## [1] FALSE
```

We obtain again the full model