Package: imv (via r-universe)

May 12, 2026

Title Model Comparison via the 'InterModel Vigorish' ('IMV')

Version 0.3

Description Computes the 'InterModel Vigorish' ('IMV'), a metric for comparing the predictive accuracy of two models for binary outcomes. The 'IMV' is derived from the expected value of a bettor using one model's predicted probabilities against those of a competing model, and is estimated via k-fold cross-validation. Methods are provided for generalized linear models, mixed-effects models ('lme4'), and item response theory models ('mirt'). See Domingue et al. (2025) <doi:10.1371/journal.pone.0316491>.

Depends R (>= 3.5.0)

Suggests lme4, mirt, parsnip, ranger, testthat (>= 3.0.0), workflows

License MIT + file LICENSE

RoxygenNote 7.3.1

Language en-US

Repository https://ben-domingue.r-universe.dev

Date/Publication 2026-05-11 21:13:05 UTC

RemoteUrl https://github.com/ben-domingue/imv

RemoteRef HEAD

RemoteSha 4af3fb05c65038a060c75ebae68ea9d424a42ee6

Contents

`imv` *Cross-validated IMV for comparing two models*

Description

S3 generic that computes the InterModel Vigorish (IMV) between a baseline model `m0` and an enhanced model `m1` via k-fold cross-validation. For each fold, both models are refit on the training partition and evaluated on the held-out partition; the IMV is computed from those out-of-fold predictions.

`imv.default` provides an escape hatch for unsupported model types via `predict_fn`: in that case the original fitted models are used (without refitting) to obtain predictions on each test fold.

Usage

```
imv(m0, m1, ...)

## S3 method for class 'glm'
imv(m0, m1, data = NULL, nfold = 4, predict_fn = NULL, y = NULL, ...)

## Default S3 method:
imv(m0, m1, data = NULL, nfold = 4, predict_fn = NULL, y = NULL, ...)
```

Arguments

`m0`	Baseline model.
`m1`	Enhanced model.
`data`	Data frame used for cross-validation. May be omitted for model classes that store training data internally (e.g., objects from `glm()` or `lme4::glmer()`).
`nfold`	Number of cross-validation folds (default 4).
`predict_fn`	Optional function with signature `function(model, newdata)` returning a numeric vector of predicted probabilities. When supplied, `imv.default` is invoked and models are not refit per fold.
`y`	Character string naming the binary outcome column in `data`. Required when `predict_fn` is supplied; otherwise inferred from the model formula.
`...`	Additional arguments passed to methods.

Value

A named list with four elements:

folds Numeric vector of per-fold IMV values (length `nfold`).

mean Mean IMV across folds.

sd Standard deviation of per-fold IMVs.

ci Named numeric vector of length 2: a 95% interval computed as mean +/- 1.96 * (sd / sqrt(nfold)).

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491

See Also

`imv.binary`, `imv.glmerMod`, `imv.SingleGroupClass`

Examples

```
## --- glm -------------------------------------------------------------
set.seed(1)
x  <- rnorm(100)
y  <- rbinom(100, 1, plogis(x))
df <- data.frame(x = x, y = y)
m0 <- glm(y ~ 1, df, family = "binomial")
m1 <- glm(y ~ x, df, family = "binomial")
result <- imv(m0, m1, nfold = 2)
result$mean
result$ci

## --- custom predict_fn (escape hatch for unsupported model types) ---
pfn    <- function(model, newdata) predict(model, newdata, type = "response")
result <- imv(m0, m1, data = df, y = "y", predict_fn = pfn, nfold = 2)
result$mean


## --- glmer (requires lme4) ------------------------------------------
if (requireNamespace("lme4", quietly = TRUE)) {
  data(sleepstudy, package = "lme4")
  sleepstudy$slow <- as.integer(sleepstudy$Reaction > 300)
  m0 <- lme4::glmer(slow ~ (1 | Subject),        sleepstudy, family = binomial)
  m1 <- lme4::glmer(slow ~ Days + (1 | Subject), sleepstudy, family = binomial)
  imv(m0, m1)
}

## --- mirt (requires mirt) -------------------------------------------
if (requireNamespace("mirt", quietly = TRUE)) {
  resp <- mirt::expand.table(mirt::LSAT7)
  mod1 <- mirt::mirt(resp, 1, "Rasch", verbose = FALSE)  # 1PL
  mod2 <- mirt::mirt(resp, 1, verbose = FALSE)            # 2PL
  imv(mod1, mod2)
}


## --- ranger (requires ranger) ---------------------------------------
## Not run:
library(ranger)
set.seed(1)
x1    <- rnorm(200); x2 <- rnorm(200)
```

```
y_fac <- factor(rbinom(200, 1, plogis(2 * x1)))
df_r  <- data.frame(y = y_fac, x1 = x1, x2 = x2)
m0 <- ranger(dependent.variable.name = "y", data = df_r[, c("y", "x2")],
             num.trees = 100, probability = TRUE)
m1 <- ranger(dependent.variable.name = "y", data = df_r,
             num.trees = 100, probability = TRUE)
imv(m0, m1, data = df_r)

## End(Not run)

## --- tidymodels workflow (requires workflows + parsnip) -------------
## Not run:
library(workflows)
library(parsnip)
set.seed(1)
x     <- rnorm(200)
y_fac <- factor(rbinom(200, 1, plogis(x)))
df_w  <- data.frame(x = x, y = y_fac)
spec <- logistic_reg() |> set_engine("glm")
wf0  <- workflow() |> add_model(spec) |> add_formula(y ~ 1) |> fit(df_w)
wf1  <- workflow() |> add_model(spec) |> add_formula(y ~ x) |> fit(df_w)
imv(wf0, wf1, data = df_w)

## End(Not run)
```

`imv.binary` *Compute IMV for binary outcomes*

Description

Computes the InterModel Vigorish (IMV) comparing baseline predictions `p1` to enhanced predictions `p2` for binary outcomes `y`. A positive value indicates that `p2` predicts better than `p1` out of sample; a negative value indicates the reverse.

Usage

```
## S3 method for class 'binary'
imv(m0, m1, p2, sigma = 1e-04, ...)
```

Arguments

`m0`	Integer or numeric vector of binary outcomes (0/1), preferably from a held-out test set.
`m1`	Numeric vector of baseline predicted probabilities (same length as `m0`).
`p2`	Numeric vector of enhanced predicted probabilities (same length as `m0`).
`sigma`	Small positive constant used to clip probabilities away from 0 and 1 to avoid numerical issues. Default `1e-4`.
`...`	Currently unused. Accepted for consistency with the `imv` generic.

Value

A scalar IMV value. Positive values favour `p2`; negative values favour `p1`.

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`

Examples

```
set.seed(1)
x  <- rnorm(1000)
y  <- rbinom(length(x), 1, plogis(x))
df <- data.frame(x = x, y = y)
m  <- glm(y ~ x, df, family = "binomial")
pr <- predict(m, data.frame(x = x), type = "response")
imv.binary(y, mean(y), pr)
```

`imv.glmerMod` *Cross-validated IMV for binomial mixed-effects models*

Description

Computes the InterModel Vigorish (IMV) for binomial mixed-effects models fit with `lme4::glmer()` via k-fold cross-validation. Both models are refit on each training fold; predictions on the held-out fold use `allow.new.levels = TRUE` to handle random-effect levels not seen during training.

Only binomial family models are supported.

Usage

```
## S3 method for class 'glmerMod'
imv(m0, m1, data = NULL, nfold = 4, predict_fn = NULL, y = NULL, ...)
```

Arguments

`m0`	A `glmerMod` object (binomial family) serving as the baseline model.
`m1`	A `glmerMod` object (binomial family) serving as the enhanced model. Must be fit to the same data as `m0`.
`data`	Optional data frame. If `NULL`, extracted from `model.frame(m1)`.
`nfold`	Number of cross-validation folds. Default 4.

`predict_fn`	Ignored for this method.
`y`	Ignored for this method; the outcome is inferred from the model formula.
`...`	Currently unused. Accepted for consistency with the generic.

Value

A named list with four elements:

- **folds** Numeric vector of per-fold IMV values (length `nfold`).
- **mean** Mean IMV across folds.
- **sd** Standard deviation of per-fold IMVs.
- **ci** Named numeric vector of length 2: a 95% interval computed as mean +/- 1.96 * (sd / sqrt(nfold)).

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`, `imv.binary`

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {
  data(sleepstudy, package = "lme4")
  sleepstudy$slow <- as.integer(sleepstudy$Reaction > 300)

  m0 <- lme4::glmer(slow ~ (1 | Subject), sleepstudy, family = binomial)
  m1 <- lme4::glmer(slow ~ Days + (1 | Subject), sleepstudy, family = binomial)
  imv(m0, m1)
}
```

`imv.ranger` *Cross-validated IMV for ranger random forests*

Description

Computes the InterModel Vigorish (IMV) for probability random forests fit with `ranger::ranger()` via k-fold cross-validation. Both models are refit on each training fold and evaluated on the held-out fold.

Models must be fit with `probability = TRUE` and a binary outcome coded as a two-level factor. The positive class is taken as the second factor level (i.e., the higher value for 0/1-coded outcomes).

Usage

```
## S3 method for class 'ranger'
imv(m0, m1, data = NULL, nfold = 5, predict_fn = NULL, y = NULL, ...)
```

Arguments

`m0`	A `ranger` object (probability forest) serving as the baseline model.
`m1`	A `ranger` object (probability forest) serving as the enhanced model.
`data`	Data frame containing all variables used by `m0` and `m1`. Required: ranger models do not store training data internally.
`nfold`	Number of cross-validation folds. Default 5.
`predict_fn`	Ignored for this method.
`y`	Ignored for this method; the outcome is taken from `m1$dependent.variable.name`.
`...`	Currently unused. Accepted for consistency with the generic.

Value

A named list with four elements:

folds Numeric vector of per-fold IMV values (length `nfold`).

mean Mean IMV across folds.

sd Standard deviation of per-fold IMVs.

ci Named numeric vector of length 2: a 95% interval computed as mean +/- 1.96 * (sd / sqrt(nfold)).

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`, `imv.binary`

Examples

```
## Not run:
library(ranger)
set.seed(1)
n  <- 300
x1 <- rnorm(n); x2 <- rnorm(n)
y  <- factor(rbinom(n, 1, plogis(2 * x1)))
df <- data.frame(y = y, x1 = x1, x2 = x2)

# m0 uses only the noise predictor; m1 uses both
m0 <- ranger(dependent.variable.name = "y", data = df[, c("y", "x2")],
```

```
             num.trees = 100, probability = TRUE)
m1 <- ranger(dependent.variable.name = "y", data = df,
             num.trees = 100, probability = TRUE)
imv(m0, m1, data = df)

## End(Not run)
```

`imv.SingleGroupClass` *Cross-validated IMV for mirt IRT models*

Description

Computes the InterModel Vigorish (IMV) for item response theory models fit with the `mirt` package via response-level k-fold cross-validation. Fold splits are at the individual response level: each held-out observation is a single person-by-item pair, and ability is estimated from the remaining responses for that person.

When a single model is supplied (`m1 = NULL`), predictions from `m0` are compared to item-level prevalence rates (the null model). When two models are supplied, `m0` serves as the baseline and `m1` as the enhanced model.

Only dichotomous response models are supported.

Usage

```
## S3 method for class 'SingleGroupClass'
imv(m0, m1 = NULL, data = NULL, nfold = 5,
     predict_fn = NULL, y = NULL,
     fscores.options = list(method = "EAP"),
     whole.matrix = TRUE,
     remove.nonvarying.items = TRUE,
     remove.allNA.rows = TRUE, ...)
```

Arguments

`m0`	A `SingleGroupClass` model object returned by `mirt::mirt()`.
`m1`	An optional second `SingleGroupClass` model fit to the same data as `m0`. When `NULL`, `m0` is compared to item prevalence.
`data`	Not used for mirt models. Accepted for consistency with the generic.
`nfold`	Number of cross-validation folds (default 5).
`predict_fn`	Not used for mirt models. Accepted for consistency with the generic.
`y`	Not used for mirt models. Accepted for consistency with the generic.
`fscores.options`	Named list of additional arguments passed to `mirt::fscores()`. Default is `list(method = "EAP")`.
`whole.matrix`	Logical (default `TRUE`). When `TRUE`, fold assignment is repeated until every training partition contains all participants and all items, ensuring that models can be re-identified on each fold. Ignored when `m1 = NULL`.

`remove.nonvarying.items`	Logical (default `TRUE`). Drop items with no response variance from each training fold's response matrix.
`remove.allNA.rows`	Logical (default `TRUE`). Drop persons whose responses are entirely missing from a training fold's response matrix.
`...`	Currently unused.

Value

A named list with four elements:

folds Numeric vector of per-fold IMV values (length `nfold`).

mean Mean IMV across folds.

sd Standard deviation of per-fold IMVs.

ci Named numeric vector of length 2: a 95% interval computed as mean +/- 1.96 * (sd / sqrt(nfold)).

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`, `imv.binary`

Examples

```
if (requireNamespace("mirt", quietly = TRUE)) {
  set.seed(1)
  resp <- mirt::expand.table(mirt::LSAT7)

  # Single model vs prevalence baseline
  mod1 <- mirt::mirt(resp, 1, "Rasch", verbose = FALSE)
  imv(mod1)

  # Two models
  mod2 <- mirt::mirt(resp, 1, verbose = FALSE)
  imv(mod1, mod2)

  # Priors specified as a variable are handled correctly
  my_prior <- list(a1 = c(0, 1, 0.25, 3))
  mod3 <- mirt::mirt(resp, 1, prior.list = my_prior, verbose = FALSE)
  imv(mod3)
}
```

`imv.workflow` *Cross-validated IMV for tidymodels workflows*

Description

Computes the InterModel Vigorish (IMV) for fitted `workflow` objects (from the **workflows** package) via k-fold cross-validation. Both workflows are refit on each training fold using `workflows::fit()` and predictions are obtained with `predict(..., type = "prob")`.

Works with any **parsnip** engine that supports probability predictions (logistic regression, random forests, boosted trees, etc.). The outcome must be a binary factor; the positive class is the second factor level.

Usage

```
## S3 method for class 'workflow'
imv(m0, m1, data = NULL, nfold = 5, predict_fn = NULL, y = NULL, ...)
```

Arguments

`m0`	A fitted `workflow` object serving as the baseline model.
`m1`	A fitted `workflow` object serving as the enhanced model.
`data`	Data frame used for cross-validation. Required: workflow models do not store training data internally.
`nfold`	Number of cross-validation folds. Default 5.
`predict_fn`	Ignored for this method.
`y`	Character string naming the binary outcome column in `data`. For formula-based workflows, `y` is inferred automatically via `workflows::extract_preprocessor()`. Required for recipe-based workflows.
`...`	Currently unused. Accepted for consistency with the generic.

Value

A named list with four elements:

folds Numeric vector of per-fold IMV values (length `nfold`).

mean Mean IMV across folds.

sd Standard deviation of per-fold IMVs.

ci Named numeric vector of length 2: a 95% interval computed as mean +/- 1.96 * (sd / sqrt(nfold)).

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

imv, imv.binary

Examples

```
## Not run:
library(workflows)
library(parsnip)
set.seed(1)
n  <- 300
x1 <- rnorm(n); x2 <- rnorm(n)
y  <- factor(rbinom(n, 1, plogis(2 * x1)))
df <- data.frame(y = y, x1 = x1, x2 = x2)

spec <- logistic_reg() |> set_engine("glm")

wf0 <- workflow() |> add_model(spec) |> add_formula(y ~ x2)
wf0 <- fit(wf0, data = df)

wf1 <- workflow() |> add_model(spec) |> add_formula(y ~ x1 + x2)
wf1 <- fit(wf1, data = df)

imv(wf0, wf1, data = df)

## End(Not run)
```

`imv0glm` *IMV for a GLM compared to a prevalence baseline*

Description

Legacy function. Computes the IMV for a fitted `glm` model against a null model (intercept only) via k-fold cross-validation. Both the full and null models are refit on each training fold and evaluated on the held-out fold.

For new code, prefer `imv(m0, m1)` with an explicit null model as `m0`.

Usage

```
imv0glm(m, nfold = 5)
```

Arguments

`m`	A `glm` object fit with a binomial family. Must have been called with an explicit `data` argument.
`nfold`	Number of cross-validation folds. Default 5.

Value

A numeric vector of length `nfold` containing the per-fold IMV values.

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`, `imv.binary`

Examples

```
set.seed(1)
x  <- rnorm(1000)
y  <- rbinom(length(x), 1, plogis(x))
df <- data.frame(x = x, y = y)
m  <- glm(y ~ x, df, family = "binomial")
imv0glm(m)
```

`imvglm.rmvar` *IMV for a GLM versus the same model with one variable removed*

Description

Legacy function. Computes the IMV comparing a full `glm` model to a reduced model with `var.nm` dropped from the formula, via k-fold cross-validation. Both models are refit on each training fold and evaluated on the held-out fold.

For new code, prefer constructing both models explicitly and calling `imv(m0, m1)`.

Usage

```
imvglm.rmvar(m, nfold = 5, var.nm)
```

Arguments

`m`	A `glm` object fit with a binomial family. Must have been called with an explicit `data` argument.
`nfold`	Number of cross-validation folds. Default 5.
`var.nm`	Character string naming the variable to remove from the formula. Must match exactly the term as it appears in the original `glm` call.

Value

A numeric vector of length `nfold` containing the per-fold IMV values. The reduced model (without `var.nm`) serves as the baseline; the full model serves as the enhanced model. A positive mean indicates that `var.nm` improves out-of-sample prediction.

References

Domingue, B. W., Rahal, C., Faul, J., Freese, J., Kanopka, K., Rigos, A., Stenhaug, B., & Tripathi, A. S. (2025). The InterModel Vigorish (IMV) as a flexible and portable approach for quantifying predictive accuracy with binary outcomes. *PloS one*, 20(3), e0316491. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316491`

See Also

`imv`, `imv.binary`

Examples

```
set.seed(1)
x  <- rnorm(1000)
z  <- rnorm(1000)
y  <- rbinom(length(x), 1, plogis(x))
df <- data.frame(x = x, z = z, y = y)
m  <- glm(y ~ x + z, df, family = "binomial")
imvglm.rmvar(m, var.nm = "z")
```

Index