

Introduction to `tbnb`

Maurizio Romano Claudio Conversano Gianpaolo Zammarchi
Giulia Contu Francesco Mola

July 21, 2026

`tbnb` implements the **Threshold-Based Naive Bayes** (Tb-NB) classifier of Romano *et al.* (2024) and its iterative refinement (iTb-NB), following an idiomatic R interface to the algorithm published in the reference papers.

```
R> library(tbnb)
R> data(toy_reviews)
R> head(toy_reviews, 3)
```

	sentiment		text
1	pos	room excellent friendly spacious location bathroom	
2	pos	spacious hotel beautiful fantastic breakfast	
3	neg	bathroom terrible rude reception leaky tiny staff rough awful	

1 Fitting Tb-NB with the formula interface

```
R> fit <- itbnb(
+   sentiment ~ text,
+   data      = toy_reviews,
+   preprocess = tbnb_preprocess(language = "english"),
+   criterion  = "balanced_error",
+   K          = 5
+ )
R> fit
```

Threshold-based Naive Bayes (tbnb)

Call:

```
itbnb.formula(formula = sentiment ~ text, data = toy_reviews,
  preprocess = tbnb_preprocess(language = "english"), criterion = "balanced_error",
  K = 5)
```

```
Classes      : neg, pos (encoded as 0/1)
Vocabulary   : 50 features
Class priors  : 0.500 / 0.500
Alpha        : 1
Threshold (tau, balanced_error) : -4.1110
```

2 Summary

```
R> summary(fit)
```

```
Tb-NB / iTb-NB model summary
=====
Classes           : neg / pos
Vocabulary size    : 50
Class priors       : 0.500 / 0.500
Laplace alpha      : 1
Selection criterion : balanced_error
Best threshold tau  : -4.1110

Top 10 positive features (highest log-odds for class 1):
      word log_odds
1  friendly    3.643
2    great    3.542
3  excellent   3.436
4    tasty    3.436
5  beautiful   3.381
6    quiet    3.381
7   helpful   3.324
8    clean    3.265
9  wonderful   3.265
10 fantastic   3.204

Top 10 negative features (lowest log-odds for class 1):
      word log_odds
1    dirty   -3.643
2    leaky   -3.593
3    cold    -3.593
4   filthy   -3.542
5    bad     -3.381
6   stained   -3.324
7 unfriendly  -3.265
8    rough    -3.204
9    noisy    -3.204
10 horrible  -3.204
```

```
Cross-validated threshold per criterion:
      metric tau_best mean_at_best sd_at_best
      precision   -4.111         1         0
      recall    -18.614         1         0
      specificity  -4.111         1         0
      fpr        -4.111         0         0
      fnr       -18.614         0         0
      f1         -4.111         1         0
      mcc        -4.111         1         0
      accuracy   -4.111         1         0
      misclassification_error -4.111         0         0
      balanced_error -4.111         0         0
```

3 Predictions

`predict()` supports four output types:

```
R> head(predict(fit, newdata = toy_reviews, type = "class"))
```

```
[1] pos pos neg pos pos neg
Levels: neg pos

R> head(predict(fit, newdata = toy_reviews, type = "score"))

[1] 8.662015 8.944149 -17.239732 7.149238 7.577015 -10.289828

R> head(predict(fit, newdata = toy_reviews, type = "prob"))

[1] 9.999972e-01 9.999979e-01 1.987219e-06 9.999871e-01 9.999916e-01
[6] 2.068485e-03
```

4 Iterative refinement (iTb-NB)

```
R> fit2 <- itbnb(
+   sentiment ~ text, data = toy_reviews,
+   preprocess = tbnb_preprocess(language = "english"),
+   criterion = "balanced_error", K = 5,
+   iterative = TRUE, iter_mode = "kde",
+   p_iter = 0.2, s_iter = 20
+ )
R> fit2$decisions
```

	iteration	start	end	tau	x_max_pos	x_max_neg	direction
1	1	-9.521677	9.478153	3.842032	9.107941	-8.97191	r

The fitted `$decisions` table can be used at predict-time when `iterative = TRUE` (the default).

5 Matrix interface

If you already have a Bag-of-Words matrix (e.g. a `quanteda::dfm` or a sparse `Matrix`), pass it directly:

```
R> fit <- itbnb(x = my_dfm, y = sentiment, criterion = "f1")
```

6 Optional embedding extension

Pass a `tbnb_embedding()` configuration to enrich the BoW with semantic neighbours of every token (an approach not present in the original papers):

```
R> emb <- tbnb_embedding(my_glove_matrix, k = 3, min_similarity = 0.6)
R> fit <- itbnb(sentiment ~ text, data = reviews, embedding = emb)
```

References

- Romano, M., Contu, G., Mola, F., Conversano, C. (2024). Threshold-Based Naive Bayes classifier. *Advances in Data Analysis and Classification*. doi:10.1007/s11634-023-00536-8
- Romano, M., Zammarchi, G., Conversano, C. (2024). Iterative Threshold-Based Naive Bayes classifier. *Statistical Methods & Applications*. doi:10.1007/s10260-023-00721-1
- Romano, M. (2025). A p-value extension for the Threshold-Based Naive Bayes classifier. doi:10.1007/978-3-031-96736-8_41