

Package: deconvodisc (via r-universe)

October 8, 2026

Title Discrete deconvolution via smoothers

Version 1.0.116

Author M.V. Bravington

Description Deconvolves count samples from unknown discrete distros
subjected to known discrete convolution matrix

Maintainer Mark Bravington <markb2@summerinsouth.net>

Imports stats, mvbutils, mgcv, TMB

LinkingTo TMB, RcppEigen

License GPL-2

Repository <https://markbravington.r-universe.dev>

Date/Publication 2024-07-11 12:34:31 UTC

RemoteUrl <https://github.com/markbravington/deconvodisc>

RemoteRef HEAD

RemoteSha cff5a26e5060e41a736b1a5ac622a99ae69c1e12

Contents

deconvodisc	1
Index	4

deconvodisc	<i>Discrete deconvolution</i>
-------------	-------------------------------

Description

Given samples from an unknown integer-valued distribution, which comes from the convolution of an unknown underlying integer-valued distro and an arbitrary-but-known (discrete) convolution matrix, this estimates the "deconvolved" underlying distro by assuming it's smooth. Can work on multiple datasets from multiple underlying distros (but all having the same convolution matrix, and all are fitted with the same estimated smoothness parameter).

Deconvolution used to be thought of as a very hard statistical problem, but the advent of "regularized methods" (to stop the underlying distro having arbitrarily wiggly solutions) has made it pretty do-able. Here package **mgcv** is used to build the underlying smooth model; a thin layer of C code using the **TMB** package overlays it, to add the convolution step.

The (co)variance matrix of the fitted distro (separate for each dataset) is also returned. It comes from `TMB::sdreport(..., ignore.parm.uncertainty=TRUE)` which ensures that the estimates across datasets appear independent; in particular, the covariances are computed conditional on the estimated smoopar (i.e. wiggleness penalty). The overall intercepts for each dataset also appear in the "outer" parameters, and are neglected covariance-wise; I *think* that's legit for Poisson/multinomial-equivalence models, but I should really check.

To do:

- Allow arbitrary and more complicated smoothers, eg random-effect on top of general smoother. Easy enough, at least as long as there's only one smoopar.
- Separate known convolution matrix for each dataset. (But you can always call `deconvodisc` several times, once for each convmat, so that's only relevant if you desperately want to share smoopars across datasets.)

Usage

```
deconvodisc( count, convmat, silent=TRUE)
```

Arguments

count	An integer-valued vector (or matrix), of bin-counts. If it's a matrix, then each column is treated as a separate dataset, from a separate underlying distro.
convmat	A numeric matrix, of dimension $(\text{length}(\text{count}) \times p)$ where p is the length of the support of the underlying distro. Note that the latter can be shorter than the observed sample.
silent	passed to <code>TMB::MakeADFun</code> . Maybe set to <code>FALSE</code> if you are having horrible problems.

Value

A list with elements `pr_truhat`, `cov`. The former is a matrix with each column (one per dataset) being the estimated probabilities for that underlying distro. The latter is an array with the covariance matrix of the probabilities for the i -th dataset in `cov[, , i]`.

Examples

```
## See "deconvo_test.r" and "deconvo_multi_test.r"
# They are in the namespace, tho should really be in the docs folder
## Not run:
# Perhaps run them yourself manually, or via the debug package
# like this (but you won't see the comments)
# NFI why I set up the example this way!
#
if( require( debug)){
  mdrun( deconvodisc:::deconvo_multi_test.r)
```

deconvdisc

3

}

End(Not run)

Index

* **misc**

deconvodisc, [1](#)

deconvodisc, [1](#)