The RJafroc Roc Book

Dev P. Chakraborty, PhD

2022-08-05

Contents

Preface

- It is intended as an online update to my "physical" book (Chakraborty, 2017).
- Since its publication in 2017 the `RJafroc` package, on which the `R` code examples in the book depend, has evolved considerably, causing many of the examples to "break".
- This also gives me the opportunity to improve on the book and include additional material.

TBA How much finished

- HMF approximately 70%
- This book is currently (as of December 2021) in preparation.
- Parts labeled TBA and TODOLAST need to be updated on final revision.

The pdf file of the book

Go here and then click on `Download` to get the `RJafrocRocBook.pdf` file.

The html version of the book

Go here to view the `html` version of the book.

A note on the online distribution mechanism of the book

- In the hard-copy version of my book (Chakraborty, 2017) the online distribution mechanism was `BitBucket`.

- BitBucket allows code sharing within a *closed* group of a few users (e.g., myself and a grad student).
- Since the purpose of open-source code is to encourage collaborations, this was, in hindsight, an unfortunate choice. Moreover, as my experience with R-packages grew, it became apparent that the vast majority of R-packages are shared on GitHub, not BitBucket.
- For these reasons I have switched to GitHub. All previous instructions pertaining to BitBucket are obsolete.
- In order to access GitHub material one needs to create a (free) GitHub account.
- Go to this link and click on Sign Up.

Structure of the book

The book is divided into parts as follows:

- Part I: Quick Start: intended for existing Windows JAFROC users who are seeking a quick-and-easy transition from Windows JAFROC to RJafroc.
- Part II: ROC paradigm: this covers the basics of the ROC paradigm
- Part III: Significance Testing: The general procedure used to determine the significance level, and associated statistics, of the observed difference in figure of merit between pairs of treatments or readers
- Part IV: FROC paradigm: TBA

Contributing to this book

I appreciate constructive feedback on this document. To do this raise an Issue on the GitHub interface. Click on the Issues tab under dpc10ster/RJafrocRocBook, then click on New issue. When done this way, contributions from users automatically become part of the GitHub documentation/history of the book.

Is this book relevant to you and what are the alternatives?

- Diagnostic imaging system evaluation
- Detection
- Detection combined with localization
- Detection combined with localization and classification
- Optimization of Artificial Intelligence (AI) algorithms

- CV
- Alternatives

ToDos TBA

- Check Bamber theorem derivation.

Chapters needing heavy edits

TBA

Shelved vs. removed vs. parked folders needing heavy edits

- replace functions with `\text{}`; eg. `\text{erf}` and `\text{exp}` in all of document
- Also for TPF, FPF etc.
- Temporarily shelved 17c-rsm-evidence.Rmd in removed folder
- Now 17-b is breaking; possibly related to changes in RJafroc: had to do with recent changes to RJafroc code - RSM_xFROC etc requiring intrinsic parameters; fixed 17-b
- parked has dependence of ROC/FROC performance on threshold

Coding aids

- sprintf("%.4f", proper formatting of numbers
- OpPtStr(, do:
- kbl(dfA, caption = "….", booktabs = TRUE, escape = FALSE) %>% collapse_rows(columns = c(1, 3), valign = "middle") %>% kable_styling(latex_options = c("basic", "scale_down", "HOLD_position"), row_label_position = "c")
- "'{r, attr.source = ".numberLines"}
- kbl(x12, caption = "Summary of optimization results using wAFROC-AUC.", booktabs = TRUE, escape = FALSE) %>% collapse_rows(columns = c(1), valign = "middle") %>% kable_styling(latex_options = c("basic", "scale_down", "HOLD_position"), row_label_position = "c")
- $\exp\left(-\lambda'\right)$ space before dollar sign generates a pdf error

- FP errors generated by GitHub actions due to undefined labels: Error: Error: pandoc version 1.12.3 or higher is required and was not found (see the help page ?rmarkdown::pandoc_available). In addition: Warning message: In verify_rstudio_version() : Please install or upgrade Pandoc to at least version 1.17.2; or if you are using RStudio, you can just install RStudio 1.0+. Execution halted

Chapter 1

Bivariate binormal model

1.1 TBA How much finished

20%

1.2 Introduction

Until now, the focus for the ROC paradigm has been on the single rating per case scenario where a single reader interprets a set of cases in a single modality. Of greater clinical interest is a study where a group of readers interprets a common set of cases in multiple modalities. This is because such a study, properly analyzed, yields results that are expected to generalize to the populations of readers and cases from which the study readers and cases were sampled. However, before one can understand several readers interpreting a common case set in multiple modalities, a good place to start is two readers interpreting a common set of cases. Historically, the field progressed from this starting point (the DBMH and ORH analyses methods came after the method to be described here). One might wonder why even bother describing the older approach, since methods for analyzing MRMC datasets have already been described. The analyses methods do not address how to fit the data to models (instead, they assume a figure of merit can be calculated and describe how to perform significance testing using the figure of merit). Data fitting is crucial to understanding factors limiting performance. Before bringing in a three-parameter model like the RSM, addressed in book Chapter 18, it is important to understand the two–parameter model applicable to paired ROC data. The analyst is dealing with two ratings per case, one per reader. An analogous situation applies when one has a single reader interpreting cases in two modalities. The analyst is again faced with two ratings per case, one per modality.

In book Chapter 06 the univariate binormal model was described, which is appropriate for modeling the single rating per case situation (the “binormal” comes from the two truth states, not the two modalities). From the previous discussion, it is evident that this needs to be extended to a bivariate model, which would describe studies where each case yields two ratings. Because of the dual interpretations the two ratings per case are correlated. Assuming a 5-rating positive-directed ROC study, on especially easy non-diseased cases the two ratings will both tend to be low (e.g., 1’s) and on especially easy diseased cases the two ratings will both tend to be high (e.g., 5’s) and on difficult cases, both ratings are likely to be in the ambiguous range (e.g., 3’s).

Here is the plan for this chapter. First, the bivariate extension of the univariate binormal model is described. Since most of the readers of this book are expected to not experts in statistics, bivariate sampling will be described at a relatively simple level. Visualization of the resulting probability density functions (pdfs) is demonstrated, which allows the reader to interactively view them, which should allow a better understand of this relatively more complex distribution. Estimation of parameters of the bivariate model from ratings data is addressed. The usage of free software implementing this procedure, i.e., CORROC2, is described. The data format used by CORROC2 is described in some detail as well as how to interpret the program output. [1] Finally, the application of the software to clinical datasets is described. A recent advance that solves degeneracy problems associated with CORROC2 is described, namely CORCBM, for correlated CBM. We conclude with a discussion of the applications of CORROC2.

1.3 Bivariate density function

Sampling from the multivariate normal distribution N_p is as follows:

$$\vec{x} \sim N_p\left(\mu, \Sigma\right) \tag{1.1}$$

Here $\vec{x}$ is a length p vector of observed samples, $\vec{\mu}$ is a length p vector containing the means of the multivariate distribution and Σ is the covariance matrix describing the variances, and correlations between the observed samples.

The multivariate normal probability density function $f_p\left(\vec{x} \;\middle|\; \vec{\mu},\ \vec{\Sigma}\right)$ is defined by:

$$f_p\left(\vec{x} \;\middle|\; \vec{\mu},\ \vec{\Sigma}\right) = \frac{1}{\left(2\pi\right)^{p/2}\left|\Sigma\right|^{1/2}} \exp\left(-\frac{\left(\vec{x}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}-\vec{\mu}\right)}{2}\right) \tag{1.2}$$

[1] Since this software runs on Windows, material is provided to allow the user to run it on a MAC (most of this necessary digression, since the author is using OS X, is in TBA Online Appendix 21.D). This digression is not needed in one is using a Windows platform.

Here $|\Sigma|$ is the determinant of the matrix and T is the transpose operator.

For the special case of a bivariate normal distribution, $p = 2$, this is equivalent to (the subscript 2 is suppressed) [2]:

$$f\left(x_1, x_2 \;\middle|\; \mu_1, \mu_2, \sigma_1, \sigma_2, \rho\right) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{t}{2\left(1-\rho^2\right)}\right) \tag{1.3}$$

where

$$t = \frac{\left(x_1-\mu_1\right)^2}{\sigma_1^2} - \frac{2\rho\left(x_1-\mu_1\right)\left(x_2-\mu_2\right)}{\sigma_1\sigma_2} + \frac{\left(x_2-\mu_2\right)^2}{\sigma_2^2} \tag{1.4}$$

For a bivariate distribution the covariance matrix is:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{1.5}$$

The `R` function to evaluate Eqn. (1.2) is `dmvnorm()`, for “density of multivariate normal distribution”, available via `R` package `mvtnorm`. Its usage is illustrated next for the following parameter values:

$$\left.\begin{aligned} \vec{\mu} &= \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 2.0 \end{pmatrix} \\ \sigma_1 &= 1.1 \\ \sigma_2 &= 1.5 \\ \rho &= 0.6 \\ \Sigma &= \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1.1^2 & 0.6 \times 1.1 \times 1.5 \\ 0.6 \times 1.1 \times 1.5 & 1.5^2 \end{pmatrix} \end{aligned}\right\} \tag{1.6}$$

In the following code the call to get the pdf occurs at line: `pdf <- dmvnorm(x, mean = c(mu1,mu2), sigma = Sigma)`.

```r
x <- c(0.1,0.2) # the x-vector at which to evaluate pdf
mu1 <- 1.5 # mean modality 1
mu2 <- 2.0 # mean modality 2
var1 <- 1.1^2  # variance modality 1
var2 <- 1.5^2 # variance modality 2
rho <- 0.6 # correlation between modalities 1 and 2
#
```

[2] In the univariate case, $p = 1$, Σ is replaced by σ^2, and the above expression reduces to the pdf of the normal distribution $N(\mu, \sigma^2)$.

```r
# construct covariance matrix Sigma
#
Sigma <- matrix(c(var1, rho*sqrt(var1*var2),
                  rho*sqrt(var1*var2), var2),2)
#
# pdf
#
pdf  <- dmvnorm(x, mean = c(mu1,mu2), sigma = Sigma)
#
# print results
#
cat("density at x1 = 0.1 and x2 = 0.2 = ", pdf, "\n")
#> density at x1 = 0.1 and x2 = 0.2 =  0.04622722
```

The parameters describing the bivariate normal distribution are $\vec{\mu}$ and Σ. The total number of parameters is five: two means, two variances and a correlation coefficient.

1.4 The bivariate binormal model

The bivariate binormal model is an extension of the bivariate model just described, to *two truth-states*, non-diseased and diseased, each modeled separately as a bivariate normal distribution. This could potentially result in a doubling of the number of parameters from 5 to 10, but by appropriate shifting and scaling transformations one can ensure that the two means of the bivariate distribution for non-diseased cases are zeroes and the two variances are each unity. This reduces the total number of parameters to 6: μ_1, μ_2, σ_1, σ_2, ρ_1, ρ_2. The bivariate binormal parameters are:

$$\left.\begin{aligned}
\overrightarrow{\mu_1} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\overrightarrow{\mu_2} &= \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \\
\overrightarrow{\Sigma_1} &= \begin{pmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{pmatrix} \\
\overrightarrow{\Sigma_2} &= \begin{pmatrix} \sigma_1^2 & \rho_2\sigma_1\sigma_2 \\ \rho_2\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}
\end{aligned}\right\} \tag{1.7}$$

The decision variable is $z_{ik_t t}$, where the i subscript corresponds to the two modalities and the t subscript corresponds to the two truth states. The correlated ratings pairs $\left(z_{1k_1 1}, z_{2k_1 1}\right)$ and $\left(z_{1k_2 2}, z_{2k_2 2}\right)$, corresponding to z-samples from non-diseased $t = 1$ and diseased cases $t = 2$, respectively, are abbreviated, using vector notation, to $\overrightarrow{z_{1k_t t}}$ (the vectorized values always "condense" the two modalities):

$$\overrightarrow{z_{k_t t}} = \begin{pmatrix} z_{1k_t t} \\ z_{2k_t t} \end{pmatrix} \tag{1.8}$$

The sampling for each modality, considered in isolation by itself, is as follows (the parameters μ and σ are of course modality dependent):

$$\left.\begin{aligned} z_{k_1 1} &\sim N(0,1) \\ z_{k_2 2} &\sim N(\mu, \sigma^2) \end{aligned}\right\} \tag{1.9}$$

Collectively $\overrightarrow{z_{k_t t}}$ is sampled from $N_2(\overrightarrow{\mu_t}, \Sigma_t)$, the bivariate normal distribution with mean $\overrightarrow{\mu_t}$ and covariance matrix Σ_t:

$$\overrightarrow{z_{k_t t}} \sim N_2(\overrightarrow{\mu_t}, \Sigma_t) \tag{1.10}$$

The parameters are truth-state dependent, Eqn. (1.7). In addition one needs to include threshold parameters in the modeling.

For an R-rating single modality ROC task with allowed ratings $r = 1, 2, ..., \mathrm{R}$, one needs $\mathrm{R} - 1$ thresholds $\zeta_1, \zeta_2, ..., \zeta_{\mathrm{R}-1}$. Defining $\zeta_0 = -\infty$ and $\zeta_{\mathrm{R}} = +\infty$ the decision rule is to label a case with rating r if the realized z-sample satisfies $\zeta_{r-1} < z \le \zeta_r$.

In the two-modality task the decision variable and the thresholds are modality dependent: i.e., $z_{ik_t t}$ and $z_{i'k_t t}$, where $i \ne i'$ and two sets of thresholds $\zeta_{i1}, \zeta_{i2}, ..., \zeta_{i(\mathrm{R}-1)}$ and $\zeta_{i'1}, \zeta_{i'2}, ..., \zeta_{i'(\mathrm{R}-1)}$. As before $\zeta_{i0} = -\infty$ and $\zeta_{i'0} = -\infty$ and $\zeta_{i\mathrm{R}} = +\infty$ and $\zeta_{i'\mathrm{R}} = +\infty$. The decision rule is to rate a case in modality i with rating r_i if $\zeta_{i(r-1)} < z_i \le \zeta_{ir}$, and the same case in modality i' is rated $r_{i'}$ if $\zeta_{i'(r-1)} < z_{i'} \le \zeta_{i'r}$.

In Eqn. (1.10), the symbol $\sim N_2(\overrightarrow{\mu_t}, \Sigma_t)$ denotes a vector sample of length two from the bivariate normal distribution with mean $\overrightarrow{\mu_t}$ and covariance matrix Σ_t. Each case yields two samples corresponding to the two modalities. The subscript 2 on N distinguishes the bivariate distribution from the corresponding univariate distribution N (which has an implicit 1 subscript). For non-diseased cases the Pearson correlation coefficient of the pairs of z-samples is ρ_1 and the corresponding value for diseased cases is ρ_2. The ones along the diagonal of Σ_1 confirm that the variances of the two samples for non-diseased cases are individually equal to unity. The corresponding variances for diseased cases are σ_1^2 and σ_2^2.

The sampling model Eqn. (1.10) is a natural extension to two modalities of the single modality univariate binormal model described in book Chapter 06. The univariate model yields one z-sample per case: samples for non-diseased cases have mean zero and unit variance; samples for diseased cases have mean μ and variance σ^2. *In contrast the bivariate model yields two z-samples per case. For non-diseased cases the samples have zero mean and unit variance and correlation ρ_1. For diseased cases, the samples for the first modality have mean μ_1 and variance σ_1^2, for the second modality they have mean μ_2 and variance σ_2^2 and their correlation is ρ_2.*

1.5 Visualizing the bivariate density function

In understanding the bivariate distribution it is helpful to visualize the pdfs. Since one is dealing with a vector $\vec{x}$ of length 2, one needs two axes to depict x_1 and x_2 and a third axis to depict the pdf. A sophisticated visualization technique is available for `R`. The package is called `plotly`. It is interactive in the sense that by dragging the cursor over the plot one can visualize it from different viewpoints. The following images were captured using this package. Shown in Fig. 1.1 are four views of the bivariate distribution density function for $\mu_{12} = 1.5$, $\mu_{22} = 2.0$, $\sigma_1 = 1.0$, $\sigma_2 = 1.5$, $\rho_1 = 0.3$ and $\rho_2 = 0.6$.

```
x <- c(0.1,0.2) # the x-vector at which to evaluate pdf
mu1 <- 1.5 # diseased mean modality 1
mu2 <- 2.0 # diseased mean modality 2
var1 <- 1  # diseased variance modality 1
var2 <- (1.5)^2 # diseased variance modality 2
rho1 <- 0.3 # non-diseased correlation
rho2 <- 0.6 # diseased correlation

#
# construction the covariance matrix Sigma1 for non-diseased
#
Sigma1 <- matrix(c(1, rho1, rho1, var2),2)

#
# pdf for non-diseased
#
# following defaults to zero mean for each modality
pdf1  <- dmvnorm(x, sigma = Sigma1)

#
# construction the covariance matrix Sigma2 for diseased
#
Sigma2 <- matrix(c(var1, rho2*sqrt(var1*var2), rho2*sqrt(var1*var2), var2),2)

#
# pdf for diseased
```

```
#
pdf2  <- dmvnorm(x, mean = c(mu1,mu2), sigma = Sigma2)

#
# print results
#
cat("mu1 =", mu1, ", mu2 =", mu2, "\n")
#> mu1 = 1.5 , mu2 = 2
cat("var1 =", var1, ", var2 =", var2, "\n")
#> var1 = 1 , var2 = 2.25
cat("rho1 =", rho1, ", rho2 =", rho2, "\n")
#> rho1 = 0.3 , rho2 = 0.6
cat("non-diseased density at x1 = 0.1 and x2 = 0.2 = ", pdf1, "\n")
#> non-diseased density at x1 = 0.1 and x2 = 0.2 =  0.1070327
cat("diseased density at x1 = 0.1 and x2 = 0.2 = ", pdf2, "\n")
#> diseased density at x1 = 0.1 and x2 = 0.2 =  0.04498399
```

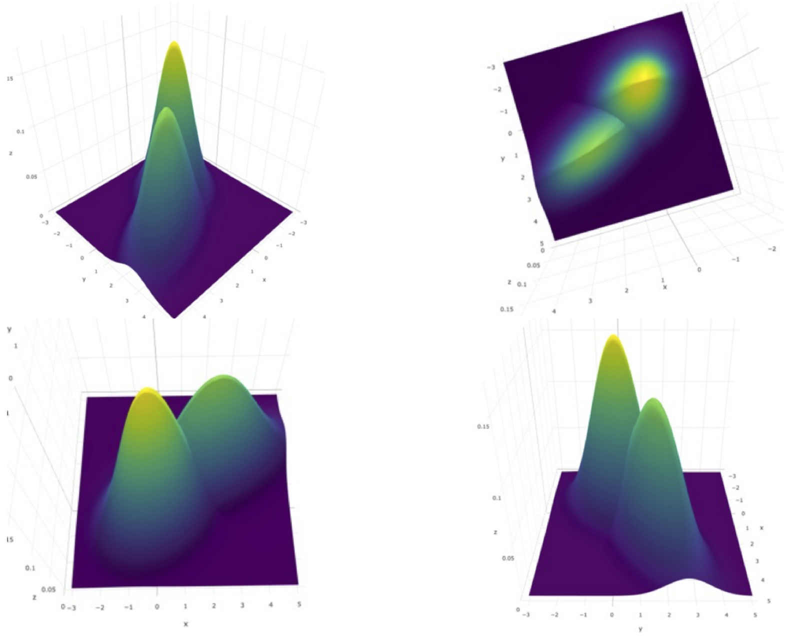

Figure 1.1: Bivariate pdf plots: Top-left is the default view. Top-right is the 'birds eye' view, with the pdf-axis 'literally coming at the reader' at right angles to the paper or screen. Bottom-left is the view looking along the y-axis at x = 0. Bottom-right is the view looking along the x-axis at y = 0. The R package plotly was used to capture these screen-shots.

1.6 Estimating bivariate binormal model parameters

In Chapter 06, a method for estimating the parameters of the univariate binormal model for a single-reader single-modality dataset was described. The method involved maximizing the likelihood function, i.e., the probability of the observed data as a function of the model parameters. Specifically, for a 5-rating study, there were four thresholds. The likelihood was a function of six variables: two basic parameters, namely the separation of the two normal distributions and the standard deviation of the diseased distribution relative to the non-diseased distribution. In addition, there are the four thresholds. The threshold parameters are sometimes referred to, in the literature, as nuisance parameters, but nuisance or not, they have to be estimated in order to get to the basic parameters. To estimate the parameters the likelihood function was maximized with respect to these parameters. The values of the parameters yielding the maximum are the maximum-likelihood estimates (MLEs).

With a bivariate model, one is dealing with six basic parameters and threshold parameters corresponding to each modality. The estimation procedure is not fundamentally different, but the presence of the second dimension does complicate the notation. Again, the starting point is the likelihood function, i.e., the probability of the observed data as a function of the parameter values. Since there is a possibility of confusion between the truth index t and the modality index i, each of which ranges from 1 to 2, the non-diseased counts in bin r of the 1st modality and bin s of the 2nd modality is denoted by (i.e., the modality indices occur before the truth index). The corresponding diseased counts are denoted . One needs two indices because each case yields two integer ratings, r and s. Each rating is in the range 1, 2,..., R, where R is the number of ROC bins. It is assumed equal for both modalities. This is not an essential limitation; the method is readily extended, and has been, for unequal numbers of bins in the two modalities. To construct the matrix one starts with a zero-initialized R x R matrix, increment the cell at row *r* and column *s* of this matrix by unity for each non-diseased case that received a r rating in the 1st modality and an s rating in the 2nd modality. When all non-diseased cases are exhausted, the count in row r and column s of this matrix is ; the procedure is repeated for the diseased cases yielding .

For non-diseased cases, the probability of a z-sample in bin r of the 1st modality and bin s of the 2nd modality is determined by the "volume" under the bivariate distribution between modality-1 thresholds and , and between modality-2 thresholds and . Complicated as this may seem, and prior to Alan Genz's work3 showing how to evaluate such multidimensional integrals, this was near impossible, but now it is quite easy, especially in R. The required probability is given by the pmvnorm() function – this is the generalization to the multivariate case of the pnorm() function, which gives the area under the normal distribution from negative infinity to a specified value. Specifically, pmvnorm()gives the probability under the multivariate distribution between specified thresholds, as illustrated next.

1.7 Examples of bivariate normal probability integrals

Needed is the bivariate extension of the univariate cumulative distribution function defined in Eqn. (3.7). Specifically, one needs a way of calculating the integral:

```
.   (21.13)
```

Online Appendix 21.C shows how to do this in R. The code file mainBivariateIntegrals.R implements the "equation":

```
.   (21.14)
```

Here `pmvnorm()` is the integral of the bivariate normal distribution with specified mean and covariance matrix sigma between limits and . The limits are specified by two length-two arrays that are supplied as the first two arguments to this function. The first argument to the function is the concatenated lower thresholds in the two modalities. The second argument is the concatenated upper thresholds in the two modalities. These are followed by the separation parameter vector and the covariance matrix. The function is not limited to two dimensional covariance matrices. With a p x p covariance matrix, the lower and upper limits would each be length p vectors. Here is a bivariate example (p = 2) in mainBivariateIntegrals.R:

1.7.1 Code listing

In the last two lines, and . Sourcing the code yields:

1.7.2 Examples of bivariate normal probability integrals

The facts that the integral over the entire space yields unity and and that over each half space yields 0.5, and , should make intuitive sense.

1.8 Likelihood function

The probability of observing and non-diseased and diseased counts, respectively, in bin r in the first modality and bin s in the second modality is:

. (21.15)

Including all bins and taking the logarithm, the logarithm of the likelihood function is given by (neglecting a combinatorial factor that does not depend on the parameters):

. (21.16)

The maximum likelihood estimates of the parameters are obtained by maximizing the function. This was solved 4,5 by Metz and colleagues more than three decades ago and implemented in software CORROC2. The Metz software is particularly useful as it measures correlations at the underlying z-sample level, which property can be used in designing a data simulator that is matched to a clinical dataset. Much as the author has emphasized that ratings are not numerical quantities, and that the observed ratings are not “hard” numbers, the measured correlations are valid, because CORROC2 models the ratings as continuous variables and estimates the correlation based on the bivariate model.

1.9 CORROC2 software

Around 1980 Prof. Charles E. Metz and Helen B. Kronman developed software implementing maximum likelihood estimation of the parameters of the bivariate normal model outlined above from data corresponding to two correlated ROC ratings per case. The software is called CORROC2, for “correlated ROC”. Subsequent revisions to the program were made by Pu-Lan Wang and Jong-Her Shen and more recently by Ben Herman4-7. CORROC2 is used to analyze paired ratings that result when a common set of cases is interpreted under two conditions. The pairing can be simple, as when a single reader interprets a common set of cases in two modalities, or more general as when two different readers interpret a common set of cases in one modality, or when two different readers interpret a common set of cases in two different modalities. The generality makes the method quite useful in a way that, in the judgment of the author, has not been fully exploited to date .

CORROC2 software allows selection of three methods of testing for differences between two conditions. 1. Testing for simultaneous differences between the (a, b) parameters corresponding to two ROC curves. 2. Testing the difference between the binormal-fitted AUCs under two correlated ROCs. 3. Testing the difference between TPFs on two ROC curves at a specified FPF.

To understand the difference between (1) and (2) consider that is that it is possible to have two ROC curves with different shapes but the same fitted AUCs. If the difference in shapes is large enough, method (1) would declare the two conditions different, but method (2) would not. In the examples below, we use method (2).

1.9.1 Details of CORROC2 software

This section is one of the few in this book that uses Windows code. If using OS X, one needs to download "virtual machine" software as described in Online Appendix 21.D. If using a Windows machine, one does not need to do anything extra. Therefore, in following this chapter it is important that one has either a Windows computer or a Mac computer with a virtual machine installed.

The Windows executable code is named CORROC2.exe. The author downloaded the software, including the Fortran source code, many years ago, ca. 1997, from the University of Chicago website, but it is currently unavailable and unsupported on that site. The author is fortunate that he held onto the downloaded code. The reason is that the FORTRAN source code, with appropriate modifications, can be interfaced to the R code. The currently distributed equivalent code named ROC-kit cannot be run in command-line mode (i.e., without having to use the mouse to select the input file, specify the output file, etc.) The author has confirmed with Mr. Ben Herman, who is familiar with the code, that by design it cannot be used in command-line mode. The ability to run in command-line mode is needed if one is to call a program from R.

For simplicity, it is assumed that the number of allowed ROC ratings bins, R, is the same in each modality (the software can handle different numbers of ratings in the two modalities). For two modalities, the total number of parameters to be estimated is 6+2(R-1). The six comes from the two means and two variances of the diseased distributions, plus the two correlations (one per truth state) and 2(R-1) from the two sets of thresholds, each with R-1 values. In addition to providing estimates of all parameters of the model, the algorithm provides estimates of the covariance matrix of the parameters. This, not to be confused with , is needed for the same reason it was needed in the univariate case, see Chapter 06: i.e., to estimate variances of the 6-estimates and variances of functions of estimates, e.g., AUC.

1.9.2 CORROC2 data input

Here is a partial example of how the paired ratings data is input to the program (the read-only data file is DataFileInp.txt in the software distribution under software/corrocii/Debug).

TYPICAL EXAMPLE OF 5-CATEGORY DATA –X 5 5 TYPICAL EXAMPLE OF 5-CATEGORY DATA –Y 5 5 4 4 3 5 5 3 1 1 1 1 1 3 1 5

The first line, whose content is immaterial to the analysis, is a description of the data in the 1st modality. The second line (5), is the total number of ratings bins in the 1st modality. The third line (5) is the rating with the highest evidence of disease in the 1st modality (5 in the current example). The fourth line is a description of data in the 2nd modality, again its content is immaterial. The next two fives are the number of bins in the second modality, and the bin number denoting highest evidence of disease in the second modality.

The actual ratings data starts on line 7, which states that the first non-diseased case was rated ROC: 4 in the first modality and ROC: 4 in the second modality. The two ratings for each case, separated by spaces, are input on the same line. Line eight states that the second non-diseased case was rated ROC: 3 in the first modality and ROC: 5 in the second modality, and so on. An asterisk after the last non-diseased case signals the end of the non-diseased cases. The next line has the two ratings corresponding to the first diseased case. Here is a partial example:

4 4 * 4 5 5 5

This says that the first diseased case was rated ROC: 4 in the 1st modality and ROC: 5 in the 2nd modality, the 2nd diseased case was rated ROC: 5 in both modalities, and so on. The last diseased case is followed by another asterisk, and the last line is the word “area”:

5 5 2 5 * area

Other choices are available for the last line, but the author will stick to area, which means one wishes to use the area under the binormal-fitted ROC curve as the figure of merit.

1.9.3 CORROC2 output

The CORROC2 output associated with this software for the above dataset, 347 lines in all, is in the read-only file DataFileOut.txt, Fig. 21.2 (use RStudio to go down the directory structure and click on the appropriate file; it is displayed, see below, with line numbers):

Fig. 21.2: Lines of CORROC2 output in DataFileOut.txt that are associated with the dataset in DataFileInp.txt.

Condition X and Condition Y denote modalities 1 and 2, respectively . Line 10 of the output summarizes the statistical test to be employed “AREA (A SUB Z) TEST”. This is because we concluded, line 101, the reader data file DataFileInp.txt with area. Line 13-22 summarizes the dataset, how many ratings, which rating represents greatest evidence of disease, etc. Lines 26 – 37 summarize the observed ratings matrix for non-diseased cases (labeled ACTUALLY NEGATIVE), i.e., the matrix . For example, there were three cases that were rated s = 3 in CONDITION X and r = 5 in CONDITION Y (the matrix is labeled by its row index, followed by its column index and last with the truth index. The output lists the 5 rating bin as the first row, the 4 rating bin as the second row, etc.). Likewise, lines 40-51 summarize the ratings matrix for diseased cases. Lines 53 – 59 list the observed operating points in the two modalities, followed by the initial and final estimates of the parameters. The program lists the parameters using the (a,b) notation, not the notation, but one knows how to transform between them: Chapter 06, Eqn. (6.16), and see exercise on conversion below. Here is the output showing the final parameter values (14 values in all):

```
            FINAL ESTIMATES OF THE PARAMETERS:
  AX= 1.3006     BX= 0.4919     AY= 1.6049     BY= 0.8704
  R(NEGATIVE CASES)= 0.2599     R(POSITIVE CASES)= 0.4544
  T(I)-0.281  0.221  0.687  1.941
  U(J)-0.791 -0.417  0.175  0.990
```

Here X refers to condition X and Y to condition Y. AX is the parameter in condition X (what the author calls). BX is the b parameter in condition X (our). AY is the parameter in condition Y (i.e.,). BY is the b parameter in condition Y (i.e.,). R(NEGATIVE CASES) is the correlation coefficient for the non-diseased cases (i.e.,). R(POSITIVE CASES) is the correlation coefficient for the diseased cases (i.e.,). Finally, T(I) and U(J) are the thresholds in the two condition, corresponding to and in our notation. Since there are five ratings bins, there are four thresholds in each modality.

Converting the above parameter values to μ, σ notation

Using the transformations Eqn. (6.16), = 1.3006/0.4919 = 2.644, = 1.844, = 1/.4919 = 2.033, = 1.149, = 0.2599 and = 0.4544 (the correlations are unaffected by linear transformations). Regarding the thresholds, for modality 1, = -0.281/.4919 = -0.571, = 0.221/.4919 = 0.449, etc. Similarly, the thresholds for modality 2 are = -0.791/.8704 = -0.909, = -0.417/.8704 = -0.479, etc. Question for the reader: why were the thresholds divided by the corresponding b-values?

1.9.4 Covariance matrix

For convenience of the yet-to-be-described R-program that does all this automatically, the FORTRAN source code was modified to print the elements of the covariance matrix sequentially, i.e., one-per-line. The output corresponding to lines 83 - 280 in file DataFileOut.txt is (shown is a partial listing):

VARIANCE-COVARIANCE MATRIX 14 0.621003E-01 0.205564E-01 0.171881E-01 0.294564E-02 0.104652E-03 -0.684784E-02 0.144067E-01 0.135140E-01 0.127711E-01 0.393710E-02 0.285687E-02 0.285592E-02 0.285248E-02 0.284967E-02 0.205564E-01

The “14” is the dimension of each “edge” of the 2D covariance matrix, i.e., there are 142 (196) elements in all. The “14” comes from the six basic parameters plus the four thresholds in the two conditions. The elements are listed in row-major order, i.e., all the elements of the 1st row, followed by all elements of the 2nd row, etc. One can anticipate some difficulty displaying it, so only the upper-left 6 x 6 corner of the full matrix is shown below, Table 21.1, i.e. any covariance involving a threshold is not shown. For convenience, the numbers have been rounded to fit and since the matrix is symmetric, there is no need to show numbers below the diagonal:

Table 21.1: This table displays the covariance matrix of the parameters for the dataset in datafileinp.txt. Since the matrix is symmetric, values below the diagonal are not shown. Also not shown are covariance-elements associated with threshold parameters.

.062 .021 .017 .003 .000 -.007

```
.022    .002    .002    .001    -.004

    .089    .045    .000    -.010

        .078    .003    -.008

            .026    .000

                .035
```

The diagonal elements, which represent variances, are all positive.

1.9.5 Rest of CORROC2 output

Line 286-320 allows one to plot the ROC curves for the two modalities, but one does not need them, as one knows by now how to generate binormal model predicted curves. Lines 325-327 lists the areas under the respective fitted ROC curves, and their correlation (the is the AUC level correlation, which could be estimated by methods described in Chapter 06):

```
        AREA(X)=  0.8784     AREA(Y)=  0.8870
        STD DEV OF AREA(X)=  0.0392     STD DEV OF AREA(Y)=  0.0376
        CORRELATION OF AREA(X) AND AREA(Y) = 0.2569
```

The remaining lines, 333-341 lists the p-value.

```
         STATISTICAL SIGNIFICANCE OF THE DIFFERENCE
  BETWEEN THE TWO CORRELATED ROC CURVES ACCORDING TO THE SELECTED TEST:
```

```
THE COMPUTED CORRELATED 'AREA TEST' STATISTIC VALUE IS   -0.1829
WITH A CORRESPONDING TWO-TAILED P-LEVEL OF 0.8549
AND ONE-TAILED P-LEVEL OF 0.4274.
```

The famed p-value is 0.8549 (it is customary to quote the two-tailed value, see Chapter 08 for details on this). This means that the difference in areas (0.8784 – 0.8870 = -0.0086) is not statistically significant at Type-I error probability = 0.05. The calculation of the test statistic is described in code mainDiffAzTest-Statistic.R. It uses methods described in Chapter 06 to calculate the standard deviation of the difference of two AUCs:

```
.   (21.17)
```

The test – statistic for determining the significance of the AUC difference is:

```
.   (21.18)
```

1.10 Application to a real dataset

Assuming one has set up the connection between the Mac side and the Windows side properly, as described in Online Appendix 21.D, one can apply CORROC2 to real ROC datasets. Open the R file "mainCorroc2.R" in the software folder corresponding to this chapter, described in detail in Online Appendix 21.E. Source the file, yielding:

1.10.1 Code Output

> source('~/book2/A ROC analysis/A8 Bivariate binormal model/software/mainCorroc2.R') …

The 6 parameters are 1.72 0.55 1.87 0.527 0.332 0.798 The 2 sided pValue is 0.482 The covariance matrix is: [,1] [,2] [,3] [,4] [,5] [,6] [1,] 0.120075 0.05096 0.050939 0.01054 6.58e-04 -1.05e-02 [2,] 0.050956 0.03667 0.013392 0.00550 2.01e-03 -4.87e-03 [3,] 0.050939 0.01339 0.111921 0.04078 2.28e-04 -1.16e-02 [4,] 0.010545 0.00550 0.040777 0.02908 1.51e-03 -4.56e-03 [5,] 0.000658 0.00201 0.000228 0.00151 2.09e-02 3.11e-06 [6,] -0.010453 -0.00487 -0.011566 -0.00456 3.11e-06 8.67e-03

What the R code does is detailed in Online Appendix 21.D. It reads the file containing the data, the Van Dyke data in the above example, extract the ratings for reader 1 in both modalities, writes the data in the appropriate format for CORROC2, runs the latter program in a Windows environment, reads the resulting output and prints out model parameters and the covariance matrix.

Like most ROC software, the output uses the (a,b) convention, Chapter 06. Recall that in this convention the diseased distribution has unit standard deviation, the non-diseased distribution has standard deviation b, and the separation of the two means is a. Since there are two modalities, there are two a parameters and two b parameters. In the CORROC2 output, the two modalities are referred to as X (1 in our notation) and Y (2 in our notation), respectively, and the ordering of the six parameters is . The ordering also determines the ordering of the covariance matrix. We start with the meanings of the header row and first column of the matrix shown above. The notation [,1] means the first column of the matrix, likewise [,2] is the second column and so on. Likewise, [1,] means the first row of the matrix, [2,] means the second row of the matrix, and so on. For example, the above table tells us that the second row and third column of the covariance matrix is 0.013392. This is the covariance between . The diagonal elements of the matrix are the variances. For example, the variance of is 2.09e-02, whose square root is the standard deviation, equal to 0.145.

In the Files panel find the folder named corrocii and open it (you need to click on the name, not the picture of the folder; watch how the folder name is underlined when the cursor is above it, which is RStudio's way of telling you "yes, it is OK to click here, something will happen"). You should see a number of files with the .FOR extension (these are FORTRAN source code files) and a folder named Debug; click on Debug. Fig. 21.4 is a screen-shot showing the contents of the Debug window. Table 21.2 summarizes the meanings of the files listed in this figure.

1.10.2 Contents of the Debug window, as viewed from the Mac side

Table 21.2: This table summarizes the meanings of the files listed in this Fig. 21.4. File name Meaning Comment DataFileInp.txt Input Ratings data in CORROC2 format Read-only file used to illustrate data input format DataFileOut.txt Output file corresponding to DataFileInp.txt Read-only file used to illustrate the output 1R2MData.txt Ratings data in CORROC2 format Created by RunCorrocOnPairedData.R, line 25 CORROC2.bat Batch file that runs CORROC2.exe See below CORROC2.exe The actual CORROC2 program
CorrocIIinput.txt Input file to the batch file
CorrocIIoutput.txt Output file created by batch file

The Windows software is located in software/CORROC2/Debug, Fig. 21.4. CORROC2.bat is a batch file, whose listing follows: 21.7.2: Contents of CORROC2.bat cd CORROCII cd Debug del CorrocIIoutput.txt CORROC2.exe < CorrocIIinput.txt > CorrocIIoutput.txt cd .. cd ..

The batch file runs on the Windows side. The first two lines change the current directory to corrocii/Debug. The next line deletes the output file, CorrocIIoutput.txt, if it exists. The next line “pipes” the contents of CorrocIIinput.txt to the executable file CORROC2.exe and “pipes” the output to CorrocIIoutput.txt. The last two lines restore the starting directory. The contents of CorrocIIinput.txt are:

1.10.3 Contents of CorrocIIinput.txt

y 1R2MData.txt N Q

These answer the four questions asked by CORROC2. Try running CORROC2 from the Windows side, Fig. 21.5.

Fig. 21.5: Result of running CORROC2 at the windows 8 command prompt.

The first line of this file is the answer (yes) to the first question asked by CORROC2.exe (you can check these out by clicking on CORROC2.exe in your physical Windows machine or virtual Windows machine): “Do you want to use data from a previously created input file for the next run?” The second line is the name of the input file, the third line is the answer (no) to the next question: “Do you want to save an output file for plotting the ROC curve?” The reason for the “no” is that we are more interested in the screen output, which contains all the estimates, including the covariance matrix, rather than plotting ROC curves. The last line is the answer (“Q”) to the next question asked by CORROC2.exe: “Do you want to use data from a previously created input file for the next run?(Y/N, or Q for quit)”. Now, in a Windows environment, click on the batch file CORROC2.bat. The latest version of the file CorrocIIoutput.txt should appear in your Explorer window, with the correct date and time stamps. Open this file. The data file is named 1R2MData.txt and if you open it, you will recognize that it is similar to the file used in the section illustrating the data input format for CORROC2. It was created by the R-code.

1.11 Discussion / Summary

CORROC2 is one of the relatively under-utilized tools developed by Prof. Charles E. Metz. There are only four publications describing it4-6,8 and two of them are difficult to find5,8, nor has the software been maintained on a level comparable to ROCFIT. For example, a method for assessing the goodness of the fit is currently not implemented. With R2 cells, it is almost impossible to maintain greater than five counts per cell, so an alternate method of combining bins has to be implemented.

One reason for the relative underutilization of CORROC2 could be that it is not designed to analyze multiple readers interpreting a common set of cases in two or more modalities i.e., MRMC datasets. For example, in a study in which CORROC2 was used9, four readers interpreted cases in four modalities. With four modalities, there are six possible combinations (4x3/2), so a conservative Bonferroni type correction for the p-value would be to divide 0.05 by 6. For each reader CORROC2 yields a p-value for the difference between the chosen pairs of modalities. If all readers agree, then there is an unambiguous answer, but the method does not allow for correlations between different readers, which undoubtedly leads to loss of power. More fundamentally, it does not address the question of interest: when averaged over all four readers, are the figures of merit in the four modalities different.

It is possible to analyze multiple reader multiple modality MRMC-ROC datasets without actually using CORROC2. In chapters Chapter 09 and Chapter 10, it was shown how one analyzes such datasets using the empirical AUCs. Alternatively, one could estimate AUC using the univariate binormal model software (e.g., ROCFIT, or more modern software). The correlations in the resulting AUC values – i.e., FOM correlations as distinct from ratings correlations - are explicitly accounted for in the DBMH / ORH analyses, i.e., it is not concerned with ratings correlations. One possible use of CORROC2 would be if one had to compare two CAD algorithms applied to a common dataset, but even here, it is of limited utility since, as a ROC study, location is ignored.

CORROC2 was developed ca. 1980 by Metz, Wang and Kronman 5,10. Subsequent revisions to the program were made by Jong-Her Shen and more recently by Benjamin Herman6,7. Its impact cannot be overstated. There are well in excess of 116 citations to this software. One reason is that, at that time, it was the only software allowing analysis of paired datasets. However, no advances have been made in the intervening 3 decades, which would allow fitting, for example, proper ROC curves to paired - and possibly degenerate - datasets. Part of the reason for this neglect is the shift in emphasis to empirical AUC based analysis, which does not require parametric modeling or curve fitting – one simply calculates the trapezoidal AUC. However, empirical AUC based analysis has its own limitations, and pursuing an improvement over the parametric approach that removes the current limitations of CORROC2 would, in the author's judgment, quite apart from the specific interest in developing realistic simulators, be of scientific interest.

As a final note, at the time of writing, 4/30/17, an extension of CORROC2 has been published11. It is called CORCBM, for correlated CBM. Details are in a document CORCBM.pdf in the online supplementary material directory corresponding to this chapter. The title of the publication is: "A bivariate contaminated binormal model for robust fitting of proper ROC curves to a pair of correlated, possibly degenerate, ROC datasets". It replaces the bivariate binormal model with the bivariate contaminated binormal model; hence its name CORCBM, for correlated CBM. Since CBM was designed to fit practically any single dataset, including degenerate ones, CORCBM is likewise able to fit practically any paired dataset. An application of CORCBM to calibrating a simulator to a data set containing single-modality multiple-reader ratings is described in Chapter 23.

Bibliography

Chakraborty, D. P. (2017). *Observer Performance Methods for Diagnostic Imaging: Foundations, Modeling, and Applications with R-Based Examples.* CRC Press, Boca Raton, FL.