Maximum Mean Discrepancy in High-Dimensional Biology

Frederick Boehm[1,*]

12 May, 2021

Abstract

Text of abstract

Contents

[1] University of Massachusetts Medical School

[*] Correspondence: Frederick Boehm <frederick.boehm@gmail.com>

Keywords: keyword 1; keyword 2; keyword 3

Highlights: These are the highlights.

1 Specific Aims

The volume and dimensionality of molecular biology data has exploded over the last twenty years. Following the sequencing of the human genome, technologies such as DNA sequencing, RNA sequencing, and single-cell RNA sequencing have proliferated. The new experimental designs and high dimension of modern biological data require parallel development of new statistical methods and computational tools for their analysis. Maximum mean discrepancy (MMD) is a new statistical tool with novel mathematical properties [5, 4, 3, 1]. We propose to explore its mathematical and statistical properties, to apply MMD methods in the analysis of high-dimensional biological data, and to build user-friendly R software tools that implement MMD methods.

Aim 1: Apply MMD to high-dimensional biological data

Two-sample problems arise in many areas of biomedical research. These include differential gene expression, comparability of data from different laboratories, and classification of cancers. Borgwardt et al. [1] examined performance characteristics of four multivariate two-sample tests in a collection of four biological settings. They found that MMD outperformed the other three tests. settings of

Aim 2: Create and maintain an open source R package that implements MMD methods

Current R software for MMD methods doesn't provide significance thresholds. Gretton's website hosts Matlab software that implements the methods in Gretton et al. [5], Gretton et al. [3] and Gretton et al. [4]. We'll implement these methods in R and C++ before assembling the code into a well documented, user-friendly R package [7, 2].

2 Background and Significance

MMD is a statistical method for determining whether two high-dimensional samples arise from distinct distributions. Throughout this document, we use the term "sample" to refer to measurements on a collection of subjects from a single group. In introductory statistics courses, one often studies Student's t-test [8] when discerning whether two samples come from the same distribution. Implicit to the t-test is the assumption that the two samples arise from normal distributions with a single, common variance parameter.

In high-dimensional settings, the analog of Student's t-test is Hotelling's T^2 test [6]. Like its univariate counterpart, Hotelling's test assumes a normal (multivariate joint normal) distribution of the random errors and a shared covariance matrix for the two groups. Departures from these assumptions may give rise to type I and type II errors in testing.

2.1 MMD and its Definition

To combat limitations of previous high-dimensional, two-sample tests, Gretton et al. [4] proposed a kernel-based test for the two-sample problem. Drawing on reproducing kernel Hilbert space (RKHS) theory, they formulate their test as a measure of distance between embeddings of probability measures in a RKHS. Gretton et al. [4] then write the test as a maximization over a subspace $\mathcal{F}$ of their RKHS, $\mathcal{H}$ (Equation (1)).

$$MMD[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_x f(x) - \mathbb{E}_y f(y) \right) \tag{1}$$

By embedding probability measures in a Hilbert space, and a RKHS in particular, Gretton et al. [5] have the option of using a rich set of analytic tools and theory. They lean heavily on the "kernel trick", which characterizes the "reproducing" nature of the RKHS $\mathcal{H}$. The Riesz representation theorem for Hilbert spaces states that there is a feature map, $\phi(x) = k(x, \cdot)$ such that $f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$ for any $f \in \mathcal{H}$. Gretton et al. [5] then define an embedding for probability distribution p as an element $\mu_p \in \mathcal{H}$ satisfying $\mathbb{E}_x f = \langle f, \mu_p \rangle_{\mathcal{H}}$.

2.2 MMD Properties

2.3 Borgwardt et al. [1] Results

In aiming to understand and characterize the performance of MMD-based statistical hypothesis tests, we begin by considering the striking results from Borgwardt et al. [1]. Borgwardt et al. [1] obtained two data sets from published breast cancer studies. Both data sets featured gene expression measurements on 2,166 common genes, but the measurements were obtained from two different microarray platforms. Borgwardt et al. [1] emphasized that, if there were no differences between the measurements on the two platforms, then the two data sets could be analyzed together. A joint analysis, when appropriate, would likely lead to greater statistical power to infer meaningful expression signatures for breast cancer. After scaling and centering expression measurements, they examined four tests. For the MMD-based test, they chose a Gaussian kernel with standard deviation set to 20. Borgwardt et al. [1] compared the MMD-based test to standard multivariate tests including Hotelling's T^2 test, the Friedman-Rafsky multivariate Kolmogorov-Smirnov test, and the Friedman-Rafsky Wald-Wolfowitz test.

In the first of two analyses, the study design consisted of randomly choosing 25 subjects' vectors of expression measurements from each sample (*i.e.*, each platform). That is, the gene expression vectors of length 2,166 were used. The two resulting subsamples consisted of 25 by 2166 matrices. To these pairs of subsamples, Borgwardt et al. [1] applied the four tests. They repeated the entire procedure 100 times. They recorded the number of instances in which each of the four multivariate tests rejected the null hypothesis that the two subsamples arose from the same platform. Table 1 tabulates the counts for the 100 instances of each test.

The second of two analyses had a similar study design, except that instead of choosing 25 subjects from each of two platforms, Borgwardt et al. [1] first chose one of the two platforms. They then chose two groups of 25 from the chosen platform and compared the resulting subsamples. Again, they repeated the procedure 100 times and recorded the results in Table 1.

Table 1: Microarray cross-platform comparability

platform	null_hypothesis	MMD	t_test	wolfowitz	smirnov
same	accepted	100	100	93	95
same	rejected	0	0	7	5
different	accepted	0	95	0	29
different	rejected	100	5	100	71

The results in Table 1 display perfect accuracy for the MMD-based test: it correctly identified 100 of 100 subsamples chosen from different platforms and 100 of 100 subsamples chosen from the same platform. The Friedman-Rafsky tests performed at an intermediate level, while the Hotelling's T^2 test performed poorly. Specifically, the Hotelling's T^2 test incorrectly called 95 of 100 cases when the two subsamples were chosen from different platforms.

Table 2: Cancer diagnosis

health_status	null_hypothesis	MMD	t_test	wolfowitz	smirnov
same	accepted	100	100	97	98
same	rejected	0	0	3	2
different	accepted	0	100	0	38
different	rejected	100	0	100	62

The remarkable performance of the MMD-based tests in these two high-dimensional settings deserves further study. It is from this perspective that we propose research to more completely characterize and understand the MMD-based test.

3 Approach

3.1 Approach for Aim 1

With the goal of enhancing understanding of the MMD-based test in high dimensions, we will undertake the following research plan. First, we'll identify existing R code that implements Hotelling's T^2 test and the two Friedman-Rafsky tests. By achieving **Aim 2**, we'll have R implementations of the MMD-based tests.

3.2 Approach for Aim 2

4 Colophon

This report was generated on 2021-05-12 12:35:03 using the following computational environment and dependencies:

```
#> - Session info ---------------------------------------------------------------
#>  setting  value
#>  version  R version 4.0.4 (2021-02-15)
#>  os       Ubuntu 21.04
#>  system   x86_64, linux-gnu
#>  ui       X11
#>  language (EN)
#>  collate  en_US.UTF-8
#>  ctype    en_US.UTF-8
#>  tz       America/New_York
#>  date     2021-05-12
#>
#> - Packages -------------------------------------------------------------------
#>  package     * version     date       lib source
#>  assertthat    0.2.1       2019-03-21 [1] CRAN (R 4.0.1)
#>  backports     1.2.1       2020-12-09 [1] CRAN (R 4.0.3)
#>  bookdown      0.21        2020-10-13 [1] CRAN (R 4.0.3)
#>  cachem        1.0.4       2021-02-13 [1] CRAN (R 4.0.3)
#>  callr         3.5.1       2020-10-13 [1] CRAN (R 4.0.3)
#>  checkmate     2.0.0       2020-02-06 [1] CRAN (R 4.0.1)
#>  cli           2.5.0       2021-04-26 [1] CRAN (R 4.0.4)
#>  colorspace    2.0-1       2021-05-04 [1] CRAN (R 4.0.4)
#>  commonmark    1.7         2018-12-01 [1] CRAN (R 4.0.1)
#>  crayon        1.4.1       2021-02-08 [1] CRAN (R 4.0.3)
#>  DBI           1.1.1       2021-01-15 [1] CRAN (R 4.0.3)
#>  desc          1.2.0       2018-05-01 [1] CRAN (R 4.0.1)
#>  devtools      2.3.2       2020-09-18 [1] CRAN (R 4.0.3)
#>  digest        0.6.27      2020-10-24 [1] CRAN (R 4.0.3)
#>  dplyr         1.0.6       2021-05-05 [1] CRAN (R 4.0.4)
#>  ellipsis      0.3.2       2021-04-29 [1] CRAN (R 4.0.4)
#>  evaluate      0.14        2019-05-28 [1] CRAN (R 4.0.1)
#>  fansi         0.4.2       2021-01-15 [1] CRAN (R 4.0.3)
#>  fastmap       1.1.0       2021-01-25 [1] CRAN (R 4.0.3)
#>  fs            1.5.0       2020-07-31 [1] CRAN (R 4.0.3)
#>  generics      0.1.0       2020-10-31 [1] CRAN (R 4.0.3)
#>  ggplot2       3.3.3       2020-12-30 [1] CRAN (R 4.0.3)
#>  glue          1.4.2       2020-08-27 [1] CRAN (R 4.0.3)
#>  gt            0.1.0       2021-05-11 [1] Github (rstudio/gt@eff3be7)
#>  gtable        0.3.0       2019-03-25 [1] CRAN (R 4.0.1)
#>  htmltools     0.5.1.1     2021-01-22 [1] CRAN (R 4.0.3)
#>  knitr         1.31        2021-01-27 [1] CRAN (R 4.0.3)
#>  lifecycle     1.0.0       2021-02-15 [1] CRAN (R 4.0.3)
#>  magrittr    * 2.0.1       2020-11-17 [1] CRAN (R 4.0.3)
#>  memoise       2.0.0       2021-01-26 [1] CRAN (R 4.0.3)
#>  munsell       0.5.0       2018-06-12 [1] CRAN (R 4.0.1)
#>  pillar        1.6.0       2021-04-13 [1] CRAN (R 4.0.4)
#>  pkgbuild      1.2.0       2020-12-15 [1] CRAN (R 4.0.3)
#>  pkgconfig     2.0.3       2019-09-22 [1] CRAN (R 4.0.1)
#>  pkgload       1.2.0       2021-02-23 [1] CRAN (R 4.0.3)
#>  prettyunits   1.1.1       2020-01-24 [1] CRAN (R 4.0.1)
#>  processx      3.4.5       2020-11-30 [1] CRAN (R 4.0.3)
#>  ps            1.5.0       2020-12-05 [1] CRAN (R 4.0.3)
#>  purrr         0.3.4       2020-04-17 [1] CRAN (R 4.0.1)
#>  R6            2.5.0       2020-10-28 [1] CRAN (R 4.0.3)
#>  remotes       2.2.0       2020-07-21 [1] CRAN (R 4.0.3)
#>  rlang         0.4.11.9000 2021-05-11 [1] Github (r-lib/rlang@7cd1f5c)
#>  rmarkdown     2.7         2021-02-19 [1] CRAN (R 4.0.3)
#>  rprojroot     2.0.2       2020-11-15 [1] CRAN (R 4.0.3)
#>  sass          0.3.1.9003  2021-05-11 [1] Github (rstudio/sass@6166162)
#>  scales        1.1.1       2020-05-11 [1] CRAN (R 4.0.1)
#>  sessioninfo   1.1.1       2018-11-05 [1] CRAN (R 4.0.1)
#>  stringi       1.6.1       2021-05-10 [1] CRAN (R 4.0.4)
#>  stringr       1.4.0       2019-02-10 [1] CRAN (R 4.0.1)
#>  testthat      3.0.2       2021-02-14 [1] CRAN (R 4.0.3)
#>  tibble        3.1.1       2021-04-18 [1] CRAN (R 4.0.4)
#>  tidyr         1.1.3       2021-03-03 [1] CRAN (R 4.0.3)
#>  tidyselect    1.1.1       2021-04-30 [1] CRAN (R 4.0.4)
#>  usethis       2.0.1.9000  2021-02-15 [1] Github (r-lib/usethis@aaf79d8)
#>  utf8          1.2.1       2021-03-12 [1] CRAN (R 4.0.3)
#>  vctrs         0.3.8.9000  2021-05-11 [1] Github (r-lib/vctrs@52157b7)
#>  withr         2.4.2       2021-04-18 [1] CRAN (R 4.0.4)
#>  xfun          0.22        2021-03-11 [1] CRAN (R 4.0.3)
#>  yaml          2.2.1       2020-02-01 [1] CRAN (R 4.0.1)
#>
#> [1] /home/fred/R/x86_64-pc-linux-gnu-library/4.0
#> [2] /usr/local/lib/R/site-library
#> [3] /usr/lib/R/site-library
#> [4] /usr/lib/R/library
```

The current Git commit details are:

```
#> Local:    main /home/fred/work/professional-development/mmdproject
#> Remote:   main @ origin (https://github.com/fboehm/mmdproject.git)
#> Head:     [70a7066] 2021-05-11: added text about probability distribution embeddings in RKHS
```

References

[1] Karsten M Borgwardt et al. “Integrating structured biological data by kernel maximum mean discrepancy”. In: *Bioinformatics* 22.14 (2006), e49–e57.

[2] Dirk Eddelbuettel and Romain François. “Rcpp: Seamless R and C++ Integration”. In: *Journal of Statistical Software* 40.8 (2011), pp. 1–18. DOI: 10.18637/jss.v040.i08. URL: https://www.jstatsoft.org/v40/i08/.

[3] Arthur Gretton et al. “A Fast, Consistent Kernel Two-Sample Test”. In: *NIPS*. Vol. 23. 2009, pp. 673–681.

[4] Arthur Gretton et al. “A kernel method for the two-sample-problem”. In: *Advances in neural information processing systems* 19 (2006), pp. 513–520.

[5] Arthur Gretton et al. “A kernel two-sample test”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[6] Harold Hotelling. “The Generalization of Student’s Ratio”. In: *The Annals of Mathematical Statistics* 2.3 (1931), pp. 360–378.

[7] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: https://www.R-project.org/.

[8] Student. “The probable error of a mean”. In: *Biometrika* (1908), pp. 1–25.