Multivariate Hypothesis Testing for Gaussian Data: Theory and Software

Tapas Kanungo and Robert M. Haralick

Intelligent Systems Laboratory
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
`tapas,haralick@george.ee.washington.edu`

Tech Report: ISL-TR-95-05

Software: `/ganesh4/tapas/hypo`

23 October, 1995

Abstract

Multivariate hypothesis testing plays a central role in statistical analysis, which is an integral part of computer vision and image processing. Although theory of univariate hypothesis testing and software implementations are readily available, the theory of multivariate testing is usually not available in one book, and we are not aware of software implementations of multivariate tests. In this paper we summarize various hypotheses that can be made about the population parameters of multivariate Gaussian distributions, and describe the various tests that can be conducted to reject these hypotheses. These tests have been implemented in C and we describe the interface to these functions. The theory and software have been validated by statistically testing empirical distributions against their theoretical distributions. The software will be available free.

KEYWORDS: Statistical analysis, multivariate hypothesis testing, software, software validation.

Contents

1 Introduction

Many computer vision problems can be posed as either parameter estimation problems (for example, estimate the pose of the object), or hypothesis testing problems (for example, the image represents which of the N objects in a database.) Since the input data (such as, images, or feature points) to these algorithms is noisy, the estimates produced by the algorithm are noisy. In other words, there is an inherent uncertainty associated with the results produced by any computer vision algorithm. These uncertainties are best expressed in terms of statistical distributions, and the distributions' means and covariances. Details of the theory and application of covariance propagation can be found in [Har94], and the references cited in that paper.

Usually, implementations of vision algorithms run into thousands of lines of code. Furthermore, the algorithms are based on many approximations, and numerous mathematical calculations. One way to check whether the software implementation and the theoretical calculations are correct is by providing the algorithm input data with known (controlled) statistical characteristics, which is possible since the input data can be artificially generated, and then checking if the estimated output is actually distributed as what was predicted by theoretical calculations.

Since many of the estimation problems are multidimensional, testing whether the means and covariances of the empirical distribution and predicted distribution are same is easier than testing whether or not the shapes of the two distributions are same. In this article, we summarize statistical tests for the case when the random estimates can be assumed to be multivariate Gaussian. We also describe the function interfaces to software we have implemented for conducting these tests. Although the software libraries and environments (e.g. Splus, numerical recipes) are available for conducting the tests for one-dimensional samples, we are un-aware of similar software libraries for multivariate case. In fact, most of the statistics books do not give all the five tests we have give (e.g. Koch [Koc87] does not address the fifth testing problem). A description of how the software and the theory was tested using statistical techniques is also included.

2 The hypotheses

Let $x_1, x_2, \ldots, x_n$ be a sample from a multivariate Gaussian distribution with population mean μ and population covariance Σ. That is, $x_i \in R^p$ and $x_i \sim N(\mu, \Sigma)$, where p is the dimension of the vectors x_i.

We can make various hypotheses regarding the population mean and covariance depending on what is known and what is unknown. The data x_i are then used to test whether or not the hypothesis is false. Notice that each population parameter (here we have two – μ and Σ) can be either (i) tested, or (ii) unknown and untested, (iii) or known. If a parameter is being tested, then a claim regarding its value is being made. If a parameter is unknown and untested, no claim is being made about the value of that parameter; its value is not known and therefore we cannot use it in any computation. If the value of a parameter assumed to be known, then its value is known without error and cannot be questioned or tested, just like the normality assumption is not questioned. Furthermore, when a parameter value is known, the value itself can be used in computation of test statistics for other parameters.

In general, if the distribution has q parameters, then there can be $3^q - 2^q$ tests. The reasoning is as follows. Since each parameter can be either tested, or unknown and untested, or known, the number of possibilities are 3^q. But, of these the number of combinations in which none of the parameters are tested (that is, they are either known, or unknown and untested – and so do not represent a test) is 2^q. Thus, the total number of distinct hypotheses that can be made about a sample from a q-parameter distribution is $3^q - 2^q$.

In the case when the data comes from multivariate normal distribution, $N(\mu, \Sigma)$, we have $q = 2$ and thus can have $3^2 - 2^2 = 5$ possible hypotheses. Now we describe each of the five tests when the data comes from a multivariate normal population.

H_1 : $\mu = \mu_0$, ($\Sigma = \Sigma_1$ **known.**) In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0. The population covariance Σ is assumed to be known and equal to Σ_1. Thus, no question can be asked regarding Σ_1 – the spread of the population from which the sample comes from is known without error, and can be used in the computation of any test statistic. The only thing that is unknown about the population is its mean. The data is used to reject the null hypothesis that the mean μ is actually equal to μ_0.

H_2 : $\mu = \mu_0$, (Σ **unknown, untested.**) In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0. No statement is made regarding the population covariance Σ is made and since its values is unknown, it cannot be used in any computation of a test statistic. Thus, one is concerned whether or not the location of the sample is around μ_0, the spread can be anything and we do not care about that.

H_3 : $\Sigma = \Sigma_0$, ($\mu = \mu_1$ **known.**) In this test, the question is whether or not the sample is from a Gaussian population whose covariance is Σ_0. The population mean μ is assumed to be known equal to μ_1. Thus, no question can be asked regarding μ_1 – the location of the population from which the sample comes from is known without error, and can be used in the computation of any test statistic. The only thing that is not known about the population is its covariance, and the data is used to reject any hypothesis about the population covariance.

H_4 : $\Sigma = \Sigma_0$, (μ **unknown, untested.**) In this test, the question is whether or not the sample is from a Gaussian population whose covariance is Σ_0. No statement is made regarding the mean μ and since its value is unknown, it cannot be used in computation of any test statistic. Thus, one is concerned whether or not the spread of the sample is around Σ_0, the location (mean) can be anywhere and we do not care about that.

H_5 : $\mu = \mu_0$, $\Sigma = \Sigma_0$ In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0 and covariance is Σ_0.

3 Definitions

In this section we briefly describe the terms used in the rest of the paper. If the reader is familiar with statistics, he/she should skip this section. For a lucid explanation of the basic univariate concepts please see [CB90]. A slightly more rigorous treatment of the univariate and multivariate test is given in [Arn90]. Multivariate tests are treated in great detail in [Koc87]. The most authoritative reference on multivariate statistics is [And84]. Although this book has most of the results, it is not very readable, and the results are scattered all over the book.

A *statistic* of the data $x_1, \ldots, x_n$ is any function of the data. For example, sample mean, $\bar{x}$, is a statistic, and so is the sample variance, S. The statistic need not be one-dimensional – $(\bar{x}, S)^t$ together form another statistic of the same data. A *sufficient statistic* is a statistic that contains all the information about the data; any inference regarding the underlying population can be made using just the sufficient statistic – the individual data points do not add any more information to the inference process. For example, the vector of original data $(x_1, \ldots, x_n)^t$ is a sufficient statistic – it contains all the information regarding the data. Another sufficient statistic is $(\bar{x}, S)^t$. Sufficient statistic is not unique. A *minimal sufficient statistic* is a sufficient statistic that has smallest number of entries. For example, for Gaussian data, $(\bar{x}, S)$ is the minimal sufficient statistic.

A *hypothesis* is any statement about a population parameter that is either true or false. The *null hypothesis,* H_0, and the *alternate hypothesis,* H_A, form the two complementary hypothesis in a statistical hypothesis testing problem.

A *test statistic* is just another statistic of the data that is used for testing a hypothesis. The *null distribution* is the distribution of the test statistic when the null hypothesis is true. The *alternate distribution* is the distribution of the test statistic when the alternate hypothesis is true.

There are two types of errors - mis-detection and false alarm. If the null hypothesis is true but the test procedure decides the null hypothesis to be false, it is called a *misdetection.* When the alternate hypothesis is true but the test procedure accepts the *null hypothesis,* it is called a *false alarm.* The misdetection probability of a test procedure is usually fixed by the user also referred to as the *significance level,* α, of the test. Typical value for α is 0.05.

The *power function* of a hypothesis test is a function of the population parameter θ, and value of the function $\beta(\theta)$ is equal to 1 minus the probability of false alarm. Ideally, the power function should be zero for θ where the null hypothesis is true and one for all θ where the alternate hypothesis is true. For most realistic testing problems one cannot create a test procedure with such an ideal power function. Power functions are very useful for evaluating hypothesis testing procedures. For an example where it is used for computer vision problems, see[KHB95]. A *uniformly most powerful test* is a test procedure whose power function is higher than all other test procedures.

There are many methods for creating tests and corresponding test statistics. The test statistics given in this paper were derived by maximizing the likelihood ratio. Please refer to the cited literature for the derivation.

4 Test statistics, null distributions and power

In this section we summarize all the test statistics and their distributions under true null hypothesis and, if known, their distributions under the alternate hypothesis. For a detailed discussion and derivations please refer to [And84].

In the following discussion we will use the following definitions of $\bar{x}$ and S.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^t,$$

where we have assumed that the data vectors x_i are p-dimensional and the sample size is n.

4.1 Test 1: $\mu = \mu_0$ with known $\Sigma = \Sigma_1$

Test statistic:

$$T = n(\bar{x} - \mu_0)^t \Sigma_1^{-1}(\bar{x} - \mu_0). \tag{1}$$

Distribution under null hypothesis is Chi-squared (Anderson, page 73):

$$T \sim \chi_p^2.$$

The alternate hypothesis is $H_A : \mu \neq \mu_0$; the distribution under the alternate hypothesis is non-central Chi-squared (Anderson, page 77):

$$T \sim \chi_{p,d}^2$$

where $d = n(\mu - \mu_0)^t \Sigma_1^{-1}(\mu - \mu_0)$ is the non-centrality parameter.
Reference: Anderson, pages 73, 77.

4.2 Test 2: $\mu = \mu_0$ with unknown Σ

Hotelling's Test statistic:

$$T = \frac{n(n-p)}{p(n-1)}(\bar{x} - \mu_0)^t S^{-1}(\bar{x} - \mu_0). \tag{2}$$

Distribution under null hypothesis (F):

$$T \sim F_{p,n-p}.$$

The alternate hypothesis is $H_A : \mu \neq \mu_0$; the distribution under the alternate hypothesis is non-central F:

$$T \sim F_{p,n-p,d}$$

where $d = n(\mu - \mu_0)^t \Sigma^{-1}(\mu - \mu_0)$ is the non-centrality parameter.
Reference: Anderson page 163.

4.3 Test 3: $\Sigma = \Sigma_0$ with known $\mu = \mu_1$

Let

$$C = \sum_{i=1}^{n}(x_i - \mu_1)(x_i - \mu_1)^t = (n-1)S + (\bar{x} - \mu_1)(\bar{x} - \mu_1)^t \ .$$

and

$$\lambda = (e/n)^{pn/2} |C\Sigma_0^{-1}|^{n/2} \exp(-tr(C\Sigma_0^{-1})/2) \ .$$

Test statistic:

$$T = -2\log\lambda. \tag{3}$$

Distribution under null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2}.$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 249, 434, 436.

4.4 Test 4: $\Sigma = \Sigma_0$ with unknown μ

Let

$$B = (n-1)S,$$

and

$$\lambda = (e/(n-1))^{p(n-1)/2} |B\Sigma_0^{-1}|^{(n-1)/2} \exp(-tr(B\Sigma_0^{-1})/2)$$

Test statistic:

$$T = -2\log\lambda. \tag{4}$$

Distribution under null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2}.$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 249, 434, 436.

4.5 Test 5: $\Sigma = \Sigma_0$ and $\mu = \mu_0$

Define

$$B = (n-1)S$$

and

$$\lambda = (e/n)^{pn/2} |B\Sigma_0^{-1}|^{n/2} \exp\left(-[tr(B\Sigma_0^{-1}) + n(\bar{x}-\mu_0)^t\Sigma_0^{-1}(\bar{x}-\mu_0)]/2\right).$$

Test statistic:

$$T = -2\log\lambda \tag{5}$$

Distribution under true null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2+p}$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$, and $\mu \neq \mu_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 442.

5 Validating theory and software

Two checks have to be performed. First check is that the theory is correct: the theoretically derived null distributions of the test statistics are actually correct. The second check is that the software is correct: the implementation is exactly what the theory dictates. Both the checks can be done by computing the empirical distributions and comparing them with the theoretically derived distributions. In the next subsection we describe how we empirically compute the null distributions of the five test statistics, and in the following section we describe how we use the Kolmogorov-Smirnov test to check if the empirical distribution and the theoretically-derived distributions are same.

5.1 Empirical null distributions

In order to generate the empirical null distributions we proceed as follows.

1. Choose some values for the multivariate Gaussian population parameters p, μ and Σ.

2. Generate n samples from the population.

3. Compute the value of the statistic, T, for the test you are verifying.

4. Repeat steps 2 and 3 M times to get T_i, $i = 1, \ldots, M$.

5. The empirical distribution T can be computed by computing the histogram of T_i.

We ran the above procedure for tests 1 through 5 described in the previous section and the plots empirical distributions of the corresponding test statistics are given in figures 1 through 5. The population parameters were: The dimension of data, $p = 4$ and $p = 1$; sample size, $n = 100$; population mean, $\mu = (1\ 2\ 3\ 4)^t$ for $p = 4$ and $\mu = 10$ for $p = 1$; population covariance, $\Sigma = 5I$ for $p = 4$ and $\Sigma = 5$ for $p = 1$; the number of repetitions $M = 500$. The histogram of the statistic and the theoretically derived function are shown in the figures. In the cases that the null distribution is distributed as χ^2, one can check the empirical distribution by using the fact that the mean, variance and mode of a χ^2_k random variable are k, $2k$, and $k-2$, respectively.

5.2 Kolmogorov-Smirnov tests

The Kolmogorov-Smirnov (KS) test tests whether two distributions are alike. The KS test uses the fact that the maximum absolute difference between the empirical cumulative distribution (the KS test statistic) and the theoretical cumulative distribution has a known distribution (the null distribution). For a more detailed discussion on the KS test see [PFTV90].

The Kolmogorov-Smirnov test was performed to check if the empirical distributions and the theoretical distributions were close enough. The p-value for the KS test are given in table 1. All the empirically computed null distributions passed the KS test. Thus we have confirmed that the theoretical derivations of the null distributions is correct and the software implementing the theory is also correct.

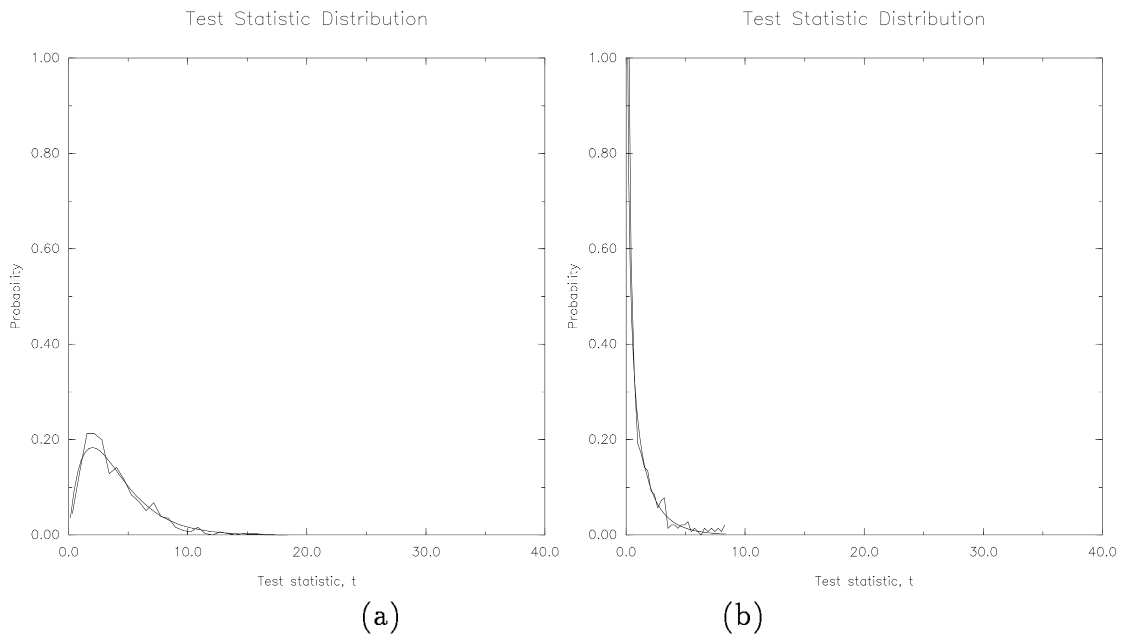
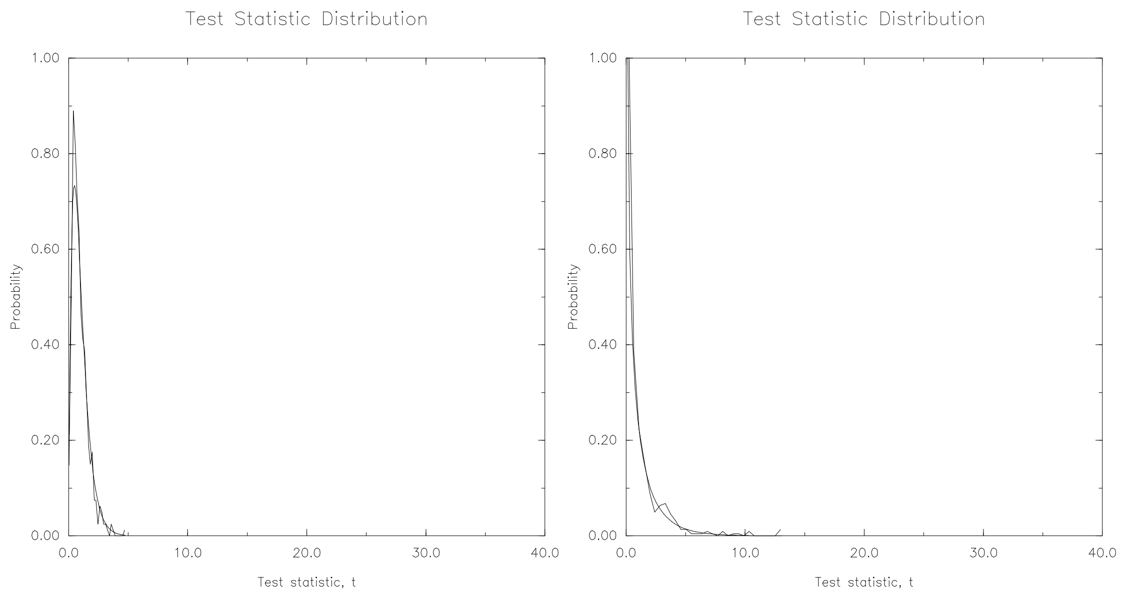
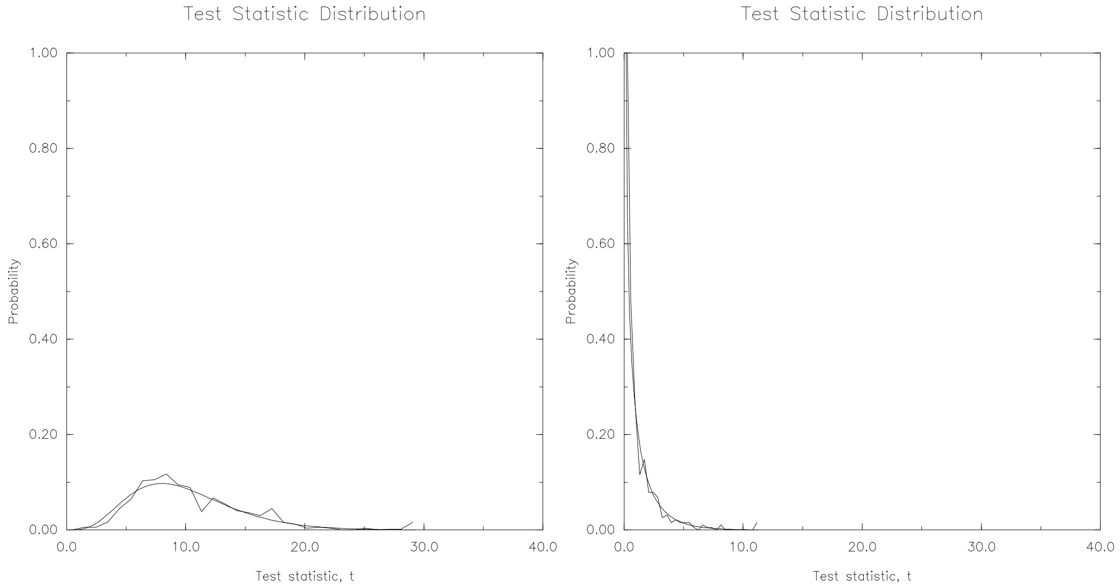


Figure 1: Emprical and theoretical distributions of the Test 1 test statistic under true null hypothesis. In this case the theoretically derived null distribution is χ_p^2 where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (1) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



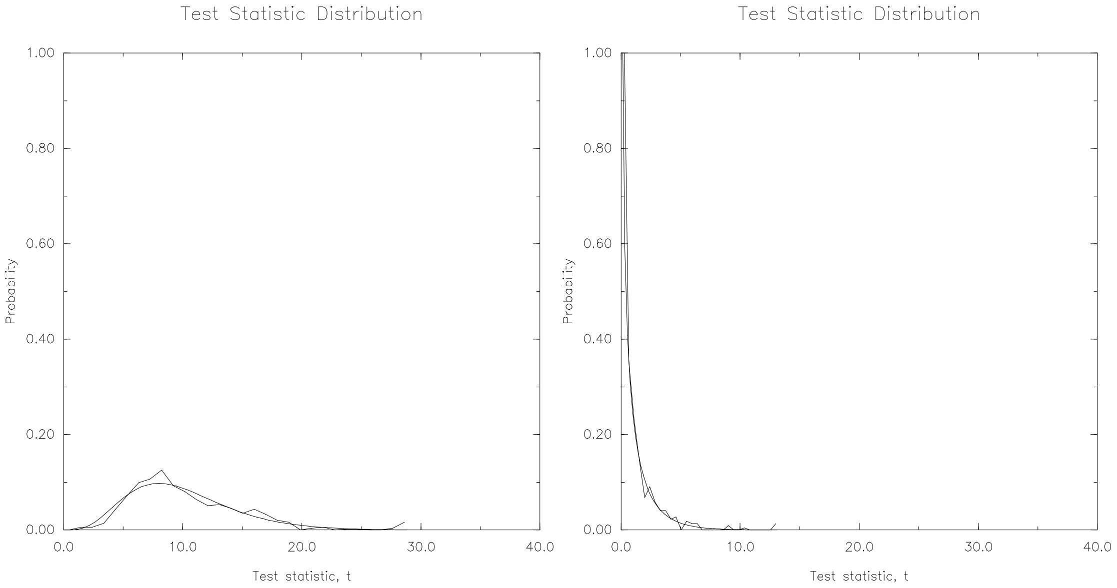
(a) (b)

Figure 2: Emprical and theoretical distributions of the Test 2 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $F_{p,n-p}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (2) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



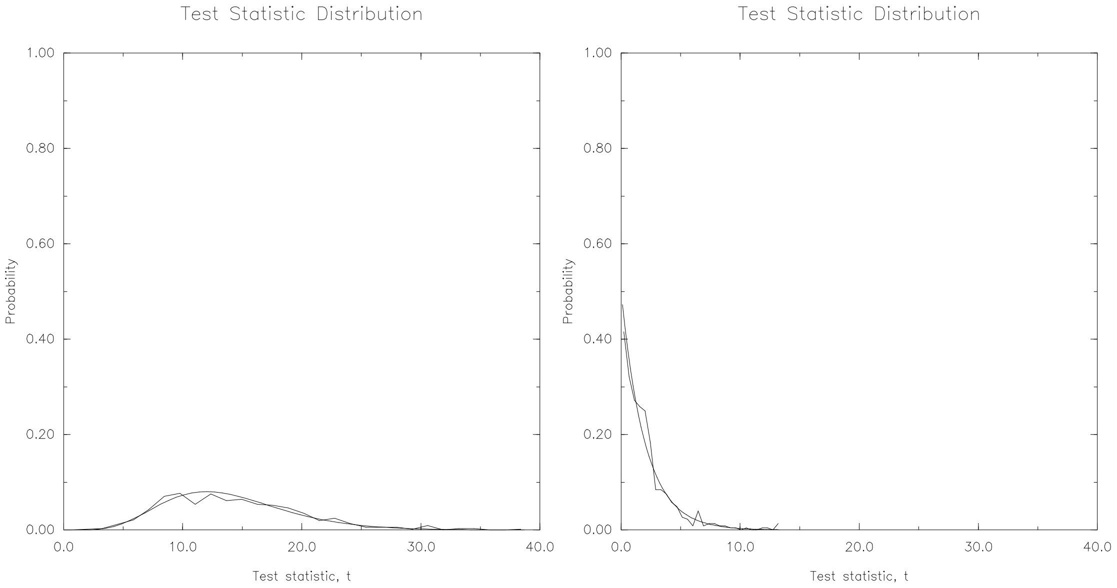
(a) (b)

Figure 3: Emprical and theoretical distributions of the Test 3 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (3) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



(a) (b)

Figure 4: Emprical and theoretical distributions of the Test 4 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (4) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



(a) (b)

Figure 5: Emprical and theoretical distributions of the Test 5 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2+p}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (5) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.

Table 1: Kolmogorov-Smirnov test results for emperical null distributions shown in figures 1 through 5. Each empirical distribution was computed using $M = 500$ test statistic values. See the text for the corresponding population parameters.

Test	Dimension p	KS P-value	Pass at α=0.05?
1	4	0.322895	Yes
1	1	0.753276	Yes
2	4	0.563564	Yes
2	1	0.820641	Yes
3	4	0.337940	Yes
3	1	0.652343	Yes
4	4	0.338493	Yes
4	1	0.827236	Yes
5	4	0.157761	Yes
5	1	0.261129	Yes

Acknowledgement

We would like to thank Professors Michael Perlman and David Madigan from the statistics department, for providing relevant literature and discussion; and Anand Bedekar and Xufei Liu for discussion.

References

[And84] T. W. Anderson. *Introduction to Multivariate Statistical Analysis.* John Wiley & Sons, Inc., New York, NY, 1984.

[Arn90] S. F. Arnold. *Mathematical Statistics.* Prentice-Hall, New Jersey, 1990.

[BC84] R. A. Becker and J. M. Chambers. *S : an interactive environment for data analysis and graphics.* Wadsworth, Belmont, CA, 1984.

[BL] B. Brown and J. Lovato. RANLIB.C: Library of C routines for random number generation. Technical report, Department of Biomathematics, The University of Texas, Houston, TX 77030. anonymous ftp: odin.mda.uth.tmc.edu:/pub/unix/ranlib.c.tar.Z.

[BLR94] B. Brown, J. Lovato, and K. Russell. DCDFLIB: A library of C routines for cumulative distribution functions, inverses, and other parameters. Technical report, Department of Biomathematics, The University of Texas, Houston, TX 77030, 1994. anonymous ftp: odin.mda.uth.tmc.edu:/pub/unix/dcdflib.c.tar.Z.

[CB90] G. Casella and R. L. Berger. *Statistical Inference.* Wadsworth, CA, 1990.

[Har94] R. M. Haralick. Propagating covariance in computer vision. In *Proc. of IAPR Int. Conf. on Pattern Recognition*, pages 493–498, Israel, October 1994.

[KHB95] T. Kanungo, R. M. Haralick, and H. S. Baird. Power functions and their use in selecting distance functions for document degradation model validation. In *Proc. of IAPR Int. Conf. on Document Anaysis and Recognition*, Montreal, Canada, August 1995.

[Koc87] K. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models.* Springer-Verlag, New York, NY, 1987.

[PFTV90] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, New York, NY, 1990.

A Function descriptions

In this section we provide descriptions of the C functions that implement the various tests described in the paper. We also used public domain software for generating random numbers (ranlib library [BL]) and for computing P-values (cdflib library [BLR94]). Few other basic routines were borrowed from [PFTV90].

The allocation of the matrices and vectors is done using the `vector()` and `matrix()` routines from Numerical Recipes (see [PFTV90]):

```
#include "utils.h"
       ...
       float    *vec;
       float    **mat;
       int      p;
        ...
       vec = vector(1,p);      /* allocate a vector */
       mat = matrix(1,p,1,p);  /* allocate a matrix */

       /* indexing with vec[i], 1 <= i <= p, is allowed */
       vec[1] = 1.0;
       vec[p] = 3.7;
       /* indexing with mat[i][j], 1 <= i,j <= p, is allowed */
       mat[1][1] = 0.0;
       mat[p][p] = 4.5;
```

NAME TestH1I()

SYNOPSIS

```
#include "mvntests.h"

void    TestH1I(n, p, barx, mu, SigmaInv, alpha, t, pval, decision)
int     n, p;
float   *barx, *mu, **SigmaInv;
float   alpha;
float   *t, *pval;
int     *decision;
```

INPUT PARAMETERS

n -- Number of samples
p -- Dimension of data vectors
barx[1..p] -- Sample mean vector
SigmaInv[1..p][1..p] -- Known population covariance matrix inverse

mu[1..p] -- Hypothesized population mean vector
alpha -- Significance level

OUTPUT PARAMETERS

t -- value of the test statistic
pval -- P-value corresponding to the test statistic
decision -- 1 if test passes, 0 if test fails

DESCRIPTION

Suppose we know that a sample of size n comes from a multivariate
distribution N(mu, Sigma) where Sigma is known. But we donot
know the population mean mu. TestH1I() uses the sample mean, xbar,
to test the hypothesis that the population mean mu = mu0, where
mu0 is any vector specified by the user.

NAME TestH2I()

SYNOPSIS

```
#include "mvntests.h"

void    TestH2I(n, p, barx, SInv, mu, alpha, t, pval, decision)
int     n, p;
float   *barx, **SInv, *mu;
float   alpha;
float   *t, *pval;
int     *decision;
```

INPUT PARAMETERS

n -- Number of samples
p -- Dimension of data vectors
barx[1..p] -- Sample mean vector
SInv[1..p][1..p] -- Sample covariance matrix inverse

mu[1..p] -- Hypothesized population mean vector
alpha -- Significance level

OUTPUT PARAMETERS

t -- value of the test statistic
pval -- P-value corresponding to the test statistic
decision -- 1 if test passes, 0 if test fails

DESCRIPTION

Suppose we know that a sample of size n comes from a multivariate
distribution N(mu, Sigma) where where both mu and Sigma are unknown.
TestH2I() uses the sample mean, xbar, and sample covariance S,
to test the hypothesis that the population mean mu = mu0, where
mu0 is any vector specified by the user.

NAME TestH3I()

SYNOPSIS

```
#include "mvntests.h"

void    TestH3I(n, p, barx, S, mu, SigmaInv, alpha, t, pval, decision)
int     n, p;
float   *barx, **S, *mu, **SigmaInv;
float   alpha;
float   *t, *pval;
int     *decision;
```

INPUT PARAMETERS

n -- Number of samples
p -- Dimension of data vectors
barx[1..p] -- Sample mean vector
S[1..p][1..p] -- Sample covariance matrix inverse
mu[1..p] -- Known population mean vector

SigmaInv[1..p][1..p] -- Hypothesized population covariance matrix inverse
alpha -- Significance level

OUTPUT PARAMETERS

t -- value of the test statistic
pval -- P-value corresponding to the test statistic
decision -- 1 if test passes, 0 if test fails

DESCRIPTION

Suppose we know that a sample of size n comes from a multivariate distribution N(mu, Sigma) where population mean mu is known. But we donot know the population covariance Sigma. TestH3I() uses the sample mean, xbar, and sample covariance, S, to test the hypothesis that the population covariance Sigma = Sigma0, where Sigma0 is any matrix specified by the user.

NAME TestH4I()

SYNOPSIS

```
#include "mvntests.h"

void    TestH4I(n, p, barx, S, SigmaInv, alpha, t, pval, decision)
int     n, p;
float   *barx, **S, **SigmaInv;
float   alpha;
float   *t, *pval;
int     *decision;
```

INPUT PARAMETERS

n -- Number of samples
p -- Dimension of data vectors
barx[1..p] -- Sample mean vector
S[1..p][1..p] -- Sample covariance matrix

SigmaInv[1..p][1..p] -- Hypothesized population covariance matrix inverse
alpha -- Significance level

OUTPUT PARAMETERS

t -- value of the test statistic
pval -- P-value corresponding to the test statistic
decision -- 1 if test passes, 0 if test fails

DESCRIPTION

Suppose we know that a sample of size n comes from a multivariate
distribution N(mu, Sigma) where Sigma is known. But we donot
know the population mean, mu, or the population covariance, Sigma.
TestH4I() uses the sample mean, xbar, and sample covariance, S,
to test the hypothesis that the population covariance Sigma = Sigma0,
where Sigma0 is any matrix specified by the user.

NAME TestH5I()

SYNOPSIS

```
#include "mvntests.h"

void    TestH5I(n, p, barx, S, mu, SigmaInv, alpha, t, pval, decision)
int     n, p;
float   *barx, **S, *mu, **SigmaInv;
float   alpha;
float   *t, *pval;
int     *decision;
```

INPUT PARAMETERS

n -- Number of samples
p -- Dimension of data vectors
barx[1..p] -- Sample mean vector
S[1..p][1..p] -- Sample covariance matrix

mu[1..p] -- Hypothesized population mean vector
SigmaInv[1..p][1..p] -- Hypothesized population covariance matrix inverse
alpha -- Significance level

OUTPUT PARAMETERS

t -- value of the test statistic
pval -- P-value corresponding to the test statistic
decision -- 1 if test passes, 0 if test fails

DESCRIPTION

Suppose we know that a sample of size n comes from a multivariate
distribution N(mu, Sigma) where Sigma is known. But we donot
know either the population mean, mu, or the population
covariance, Sigma. TestH5I() uses the sample mean, xbar,
and sample covariance, S, to test the hypothesis that the population
mean mu = mu0 and population covariance Sigma = Sigma0, where
mu0 and Sigma0 are user-specified vector and matrix, respectively.