A general method for goodness-of-fit tests for arbitrary multivariate models

Lolian Shtembari
Max Planck Institute for Physics, Munich, DE 80805

Goodness-of-fit tests are often used in data analysis to test the agreement of a model to a set of data. Out of the box tests that can target any proposed distribution model are only available in the univariate case. In this note I discuss how to build a goodness-of-fit test for arbitrary multivariate distributions or multivariate data generation models. The resulting tests perform an unbinned analysis and do not need any trials factor or look-elsewhere correction since the multivariate data can be analyzed all at once. The proposed distribution or generative model is used to transform the data to an uncorrelated space where the test is developed. Depending on the complexity of the model, it is possible to perform the transformation analytically or numerically with the help of a Normalizing Flow algorithm.

UNIVARIATE GOODNESS-OF-FIT TESTS

Assessing the goodness-of-fit of a distribution given a number of random samples is an often-encountered problem in data analysis. Several non-parametric tests exist, including the Kolgogorov-Smirnov (KS) test [1, 2], the Anderson-Darling (AD) test [3] or the Recursive Product of Spacings (RPS) test [4], some of which have become standard tools. These tests can be used to perform the goodness-of-fit of any univariate distribution. This flexibility in targeting vastly different univariate distributions is made possible by the probability integral transformation [5, 6].

Probability integral transformation

Suppose we have \( m \) samples \( \{x_i\} \), and want to quantitatively test the hypothesis of those samples being random variates of a known distribution \( f(x) \), independent and identically distributed (i.i.d.) according to \( f(x) \). Considering only continuous distributions \( f(x) \) with cumulative \( F(x) \), it is possible to transform samples onto the unit interval \([0,1]\) via \( y_i = F(x_i) \). This reduces the task at hand to test transformed samples \( \{y_i\} \) being distributed according to the standard uniform distribution \( U(0,1) \).

UNIFORMITY TEST IN THE UNIT HYPER-CUBE

The take-away message from the univariate case is that, in order to develop a test statistic, it is easier to do so in a standardized space, such as the uniform interval \([0,1]\). Much like the univariate case, the goal is to develop a uniformity test for the unit hyper-cube in \( n \) dimensions \([0,1]^n\).

Suppose we have \( m \) samples \( \{x_i\} \in [0,1]^n \). Under the assumption of a uniform distribution in each dimension, the \( n \) components of each sample are independent of one another, meaning that the \( n \) dimensions are completely uncorrelated. The projections of the samples along each axis of the hyper-cube yields \( n \) univariate uniformly distributed sets of data: \( \{x_{ij}\} \) for the \( j \)-th dimension. For each one of these projected datasets \( \{x_{ij}\} \) it is possible to perform a uniformity test using a test statistic of choice and condense the information for the \( j \)-th dimension in one scalar p-value \( p_j \). Under the assumption of a uniform distribution, the expected distribution of each p-value \( p_j \) is uniform, and moreover, each p-value will be independent of one another.

Performing the uniformity test on projections of our original samples along all dimensions, we end up with a list of \( n \) p-values \( \{p_j\} \), which are independent and identically distributed according to a uniform distribution. On this resulting dataset, \( \{p_j\} \), it is possible to run once more a uniformity test using a test statistic of choice in order to check whether there are any significant deviations from uniformity. The result of this last uniformity check results in one last p-value \( p_{final} \) which is the overall p-value of the multivariate goodness-of-fit test.

As pointed out in the discussion above, in order to obtain the intermediate p-values, \( \{p_j\} \), and then the final one, \( p_{final} \), it is possible to use any test statistic of choice, as long as the chosen statistics preserve the non-correlation among dimensions (results from tests that have a Poisson dependent factor, for example, will be correlated, since the same number of samples is projected on all dimensions). What is important is that the distribution of the resulting p-values is uniform. This means that the test statistic used for the evaluation of the intermediate p-values, \( \{p_j\} \), does not have to be the same as the one used to evaluate \( p_{final} \); as a matter of fact one could also use different tests for different dimensions in the evaluation of \( \{p_j\} \), but it might be a more consistent approach to consider all dimensions equally and use the same test for all projections.

In the previous discussion, we considered a dataset of \( m \) samples \( \{x_i\} \in [0,1]^n \). In such a case, if \( m \) is large, it might be appropriate to use a test such as RPS or KS in order to pick up on a signal in any of the projections. Afterwards, when considering the \( n \) p-values \( \{p_j\} \), it could
be better to look for outliers, since already one of a few small \( p_j \) could be indicative of the presence of a signal in our data. In this case, especially when dealing with low-dimensionality spaces \((n\text{ small})\), instead of using RPS or KS on the set \( \{p_j\} \) it might be more informative to look at the smallest p-value or rather their product in case we want to improve the sensitivity in the presence of multiple small p-values.

**Minimum p-value**

As discussed, observing one small p-value might already be enough to point to a possible signal in the data. Under the assumption of a uniform distribution of \( \{p_j\} \), the distribution of \( p_{\text{min}} = \min\{p_j\} \) is simply the first Order Statistic, and it follows a Beta distribution [7]:

\[
p_{\text{min}} = \min_j\{p_j\} \sim \text{Beta}(1,n)
\]  

where \( n \) is the dimensionality of the original data. Thus the final p-value is:

\[
p_{\text{final}} = F_{\text{Beta}}(p_{\text{min}}; 1,n)
\]

where \( F_{\text{Beta}}(x; a,b) \) is the cumulative distribution function of the Beta distribution with parameters \( (a,b) \).

**Product of p-values**

If there is more than one small p-value \( p_j \), looking only at the smallest one might be reductive while we could gain in sensitivity by combining the small p-values together. One way of doing so is to consider the product of all p-values:

\[
p_{\text{prod}} = \prod_{j=1}^n p_j
\]

Once again, we expect all \( \{p_j\} \) to be uniformly distributed, so the distribution of \( p_{\text{prod}} \) is known [8]:

\[
P(p_{\text{prod}} = x) = \frac{(-1)^{n-1}}{(n-1)!} [\ln(x)]^{n-1}
\]

thus the final p-value \( p_{\text{final}} \) is:

\[
p_{\text{final}} = F(p_{\text{prod}}) = p_{\text{prod}} \sum_{j=1}^n \frac{(-1)^{j-1}}{(j-1)!} [\ln(p_{\text{prod}})]^{j-1}
\]

**TRANSFORMING TO THE UNIFORM UNIT HYPER-CUBE**

Till now the multivariate goodness-of-fit test was developed assuming a uniform distribution in the unit hyper-cube \([0,1]^n\). In order to be able to use this method targeting any given multivariate distribution \( M \), it is imperative to be able to transform the probability space described by \( M \) into \([0,1]^n\). This transformation can be easy or difficult depending on the model \( M \), specifically depending on the correlation among the dimensions of \( M \). In the following I show how to perform the transformation into the unit hyper-cube in three main cases: first, models comprised of uncorrelated dimensions are considered, moving then to models with correlated dimensions or sample generating processes for which a probabilistic model is not available and finally hierarchical models are discussed.

**Independent dimensions**

If the dimensions of the proposed model \( M \) are all independent of each other, then \( M \) is just a composition of \( n \) independent univariate models:

\[
M = [M_1, M_2, ..., M_n]
\]

where \( M_j \) is the distribution of the \( j \)-th dimension. Much like the univariate case, it is possible to transform the \( j \)-th component of each sample using the corresponding cumulative distribution function \( F_{M_j} \). Thus, the transformation of sample \( x = (x_1, x_2, ..., x_n) \) in \([0,1]^n\) is simply:

\[
u = [u_1, u_2, ..., u_n] = [F_{M_1}(x_1), F_{M_2}(x_2), ..., F_{M_n}(x_n)]
\]

In the following, the model dependent transformation will be written as such:

\[
u = T_M(x)
\]

**Correlated dimensions and generative models**

If the dimensions of the model are not mutually independent, then it might be difficult to write down a transformation to the hyper-cube. This is still possible when dealing with nicely behaved distributions, such as a multivariate Normal distribution whose covariance matrix is not diagonal, but that might not be the case for a more complex model, such as a weighted sum of distributions. In such cases, it is possible to learn the transformation to the unit hyper-cube by using a Normalizing Flow (NF) which can perform whitening of the data, i.e. transform
the data so that they are distributed according to a Standard multivariate Normal distribution. Once the data is transformed into a set of Standard multivariate Normal variates, it is then possible to transform to the unit hyper-cube one component at a time as shown earlier.

The Normalizing Flow is made up of a neural network which is trained using samples from the proposed model \( M \). The samples needed for training can be obtained from an associated generative model or by sampling \( M \) using a Markov chain Monte Carlo. The use of the generative model is particularly interesting because it allows to train the Normalizing Flow without having a normalized model or any model at all. In such cases, the NF is learning the associated distribution model and the transformation all at once. [9, 10] offer a nice review of the many applications of Normalizing Flows. In order to show the feasibility of this approach, a proof of principle example is presented where a Normalizing flow is used to whiten data sampled from a sum of three two-dimensional Normal distributions. A sampled distribution of this model is depicted in Fig. 1 and the resulting marginal distributions of the whitened samples are shown in Fig. 2. The Normalizing Flow used for this example was adapted from [10].

![FIG. 1. Sample distribution of the sum of three two-dimensional Gauss distributions.](image1)

Hierarchical models

Given a hierarchical model, the distribution of some components of the data is dependent of the values of other components, which are referred to as hyper-parameters of the model. If the hyper-parameters are mutually independent or if a transformation to the unit hyper-cube is available for their distribution and if the same is true for all the dependent parameters at each layer of depth of the hierarchical model, then it is possible to transform the whole model into the unit hyper-cube in stages.

Consider for example the following 2 layer hierarchical model \( M = [M_1, M_2(M_1)] \). \( M_1 \) models the distribution of the hyper-parameters \( x^{high} \) of the model and these components can be transformed to the corresponding uniform unit hyper-space using the associated function \( T_{M_1} \). The distribution of the dependent parameters \( x^{low} \) is affected by the observed value of the hyper-parameters \( x^{high} \):

\[
x^{low}_i \sim M_2(x^{high})
\]  

For any given sample \( x_i \), the value of the hyper-parameters \( x^{high}_i \) is fixed, so the model \( M_2(x^{high}) \) is fully defined and it is possible to compute the corresponding transformation to the unit hyper-space. While \( T_{M_1} \) is sample-independent, \( T_{M_2} \) is sample-dependent. In case of hierarchical models with more layers, this staged transformation approach is to be repeated for each layer.

![FIG. 2. Whitened marginal distributions after transforming with the Normalizing Flow.](image2)

EXAMPLE

Here we illustrate how the proposed goodness-of-fit test could be used in a physics scenario. Assume the background of an experiment follows a known distribution, unaffected by any additional signal. After having collected some data, one wants to quantify the goodness-of-fit of the background only model to the data and a resulting low p-value could indicate the presence of events distributed according to an additional, unknown signal distribution.

In this example the background is modelled by a simple uniform distribution in the 5-dimensional hyper-cube [0, 1] \( ^5 \) and in order to illustrate how the presence of an actual signal (alternative hypothesis) would affect the outcome, additional events are injected, following a multivariate Normal distribution positioned at the center of the hyper-cube with isotropic variance of either 0.01 or 0.1. The number of events is Poisson fluctuated for both background and signal populations, with expected values of \( \langle n_b \rangle = 10^4 \) and expected values of \( \langle n_s \rangle \) ranging up to \( 10^3 \).

The p-value distributions under the assumption of \( H^0 \) (i.e. only background is present) are shown in Fig. 3. The evaluation of the intermediate p-values was performed using the KS test, given the large count rates,
FIG. 3. p-value distributions for background only samples \( \langle n_s \rangle = 0 \) and background plus randomised signal injections in 5 dimensions: "narrow" signal with \( \Sigma = 0.01 \cdot I_5 \) (top) and "wide" signal with \( \Sigma = 0.1 \cdot I_5 \) (bottom); comparison to the background model for either the minimum p-value statistic (left) or the product of p-values statistic (right).

while the evaluation of the final p-value, since there are only 5 dimensions, was performed using the two tests previously describe, namely the minimum and the product of intermediate p-values. Distributions with no signal \( \langle n_s \rangle = 0 \) show a flat p-value distribution, as expected, while the distributions of trials with injected signals are trending towards smaller p-values, indicating the worsened goodness-of-fit for the background only model. The distributions of trials where the signal has smaller variance (top) are much more skewed towards small p-value compared to those where a larger variance signal was injected (bottom). This shows how the sensitivity of the tests when targeting clusters of varying width and strength relative to the background. In this example, since the signal can be spotted in the projection of multiple dimensions, the product of p-values test (right) offers the largest rejection probability of the null hypothesis compared to the minimum p-value test (left), which can only rely on the sensitivity of one projection.

Note that the data in the previous examples were analyzed all in one pass for each trial, meaning that the extracted p-values do not need any corrections for a 'trials effect' or 'look-elsewhere effect'. Of course, if one analyzes many separate sets of data, the resulting p-value will need to be corrected as is usually done in the univariate case.

CONCLUSIONS

In this note a general method to perform goodness-of-fit tests targeting any given multivariate distribution or multivariate generative model is proposed by means of a transformation to the uniform unit hyper-cube. Furthermore, two different test statistics are proposed, namely the minimum p-value test and the product of p-values test and their sensitivity is discussed in the contest of a mock signal search. The tests developed with this method perform an unbinned analysis of the data and do not need any trials factor or look-elsewhere correction since the multivariate data is analyzed all at once.

ACKNOWLEDGEMENTS

I thank Prof. Allen Caldwell and Dr. Oliver Schulz for helpful discussions and comments. I thank Dr. Oliver Schulz, Dr. Vasyl Hafych and Michael Dudkowiak for their help in the implementation of the Normalizing Flow used in this note.

[1] A. Kolmogorov, G. Ist. Ital. 4, 83–91 (1933).
[2] N. Smirnov, Annals of Mathematical Statistics 19, 279 (1948).
[3] T. W. Anderson and D. A. Darling, Journal of the American Statistical Association 50, 765 (1954).
[4] F. Eller and L. Shtembari, (2021), 2111.02252.
[5] K. Pearson, Biometrika 1, 390 (1933).
[6] K. Pearson, Biometrika 25, 379 (1933).
[7] H. A. David and H. N. Nagaraja, Order statistics (Wiley, 2003).
[8] M. Springer, The Algebra of Random Variables Probability and Statistics Series (Wiley, 1979).
[9] I. Kobyzev, S. J. Prince, and M. A. Brubaker, IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 3964 (2021).
[10] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, (2019).