Goodness-of-fit Function Used in Python

Rui Gao (rui.gao@aggiemail.usu.edu)

1. Introduction

Such as evapotranspiration and leaf area index can be estimated via trained machine learning methods when the ground measurements are available. A variety of metrics can tell us how the performance of the machine-learning model is. Based on my current experience and considering that python is popular since it could be combined with other software such as ArcGIS, this project is aiming at providing a python function package and also basic concepts of each goodness-of-fit methods in order to provide a potential way to compare the results from different research paths and to save time to look for the details about each goodness-of-fit method. This repository is organized to show the python package in the first part, followed by the description for each method, and one example shows at the end.

The symbols used in this paper include:

$\boldsymbol{n}$**:** the number of observations

$\boldsymbol{O}$**:** it stands for the ground observation

$\boldsymbol{\bar{O}}$**:** the average of ground observations

$\boldsymbol{O_i}$**:** ground observations

$\boldsymbol{E}$**:** it stands for the estimation gained via ML methods

$\boldsymbol{\bar{E}}$**:** the average of estimations gained from ML models

$\boldsymbol{E_i}$**:** estimated value via ML models

2. Goodness-of-fit function package

Two input vectors, observations and estimations, are supposed to be provided. Currently, this function package contains 7 different goodness-of-fit statistics, including root mean square error (RMSE), relative root mean square error (RRMSE), mean absolute error (MAE), correlation coefficient (r), coefficient of determination (R^2), coefficient of efficiency (E), and mean squared error (MSE). It will be explained one by one in the next part.

```python
def gfit(true,pred,type_statistic='1',residual='Yes'):
    import numpy as np
    import pylab
    import matplotlib.pyplot as plt
    '''
    true: it supposed to be the observations
    pred: it supposed to be the estimations from (ML) models
    type_statistic: folow the information below to see which one you need
    residual: if "Yes", the residual plot will be provided, otherwise, no residual plot
    type_statistic='1': Root Mean Square Error - RMSE (default)
    type_statistic='2': Relative Root Mean Square Error - RRMSE
    type_statistic='3': Mean Absolute Error - MAE
    type_statistic='4': Correlation Coefficient - r
    type_statistic='5': Coefficient of Determination - R2
    type_statistic='6': Coefficient of Efficiency - E
    type_statistic='7': Mean Squared Error - MSE
    '''

    # Size of two vectors should be the same
    # error will come up if the size does not match
    if true.shape == pred.shape:
        pass
    else:
        print("\nError! The size of vector 1 does not match vector 2!\n")

    # Basic information about inputs
    num_element = len(true)
    diff = true - pred
    mean_true = true.mean()
    mean_pred = pred.mean()

    # residual plot (optional)
    if residual=='Yes':
        plt.figure(figsize=(5, 2.5))
        plt.scatter(range(1, 1+len(true)),diff)
        plt.plot(range(1, 1+len(true)),[0]*len(true),'r--')
        plt.xlim(0, len(true)+1)
        plt.ylabel("Residual")
        plt.show()
    else:
        pass

    # Calculate the goodness-of-fit statistics
    # RMSE (root mean square error)
    if type_statistic == '1':
        out = np.sqrt((diff**2).mean())
    # RRMSE (relative root mean square error)
    elif type_statistic == '2':
        out = np.sqrt((diff**2).mean())
        out = out/mean_true*100
    # MAE (mean absolute error)
    elif type_statistic == '3':
        out = (abs(diff)).mean()
    # r (correlation coefficient)
    elif type_statistic == '4':
        tmp = np.corrcoef(true,pred)
        out = tmp[0,1]
    # R2 (coefficient of determination)
    elif type_statistic == '5':
        tmp = np.corrcoef(true,pred)
        tmp = tmp[0,1]
        out = tmp**2
    # E (coefficient of efficiency)
    elif type_statistic == '6':
        out = 1 - (diff**2).sum()/((true - mean_true)**2).sum()
    # MSE (mean squared error)
    elif type_statistic == '7':
        out = diff**2
        out = out.mean()

    return(out)
```

Figure 1. Screenshot showing the python function

Before looking at the statistical measures for goodness-of-fit, the residual plot is presented inside the function firstly (the middle part showed in Figure 1). Because the residual plots can reveal unwanted residual patterns that indicate biased results more efficiently than numbers. When the residual plots pass muster, the numerical results can be trusted, and then we can move to check the goodness-of-fit statistics[1]. The residual plot is an option. The plot will show up when "residual = 'Yes'".

3. Goodness-of-fit statistics

1) Root mean squared error (RMSE)

Basic concepts

The RMSE is a standard statistical metric, especially in the field of geosciences (McKeen et al. 2005), to measure the performance of the models in a variety of research fields (Chai, Draxler 2014). In (Chai, Draxler 2014), it also shows that the RMSE is more appropriate to represent model performance than the MAE when the error distribution is expected to be Gaussian.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(E_i - O_i)^2}{n}}$$

RMSE is the standard deviation of the residuals (the difference between the observation and the estimation). The unit of RMSE is the same as the observation. Compared with mean absolute error, RMSE is more sensitive to outliers once the errors are squared in this equation before they are summed (Käfer and da Rocha 2020).

Python code

```
# Basic information about inputs
num_element = len(true)
diff = true - pred
mean_true = true.mean()
mean_pred = pred.mean()

# RMSE (root mean square error)
if type_statistic == '1':
    out = np.sqrt((diff**2).mean())
```

2) Relative root mean square error (RRMSE)

Basic concepts

The RRMSE is a dimensionless version of RMSE (Aboutalebi et al. 2019). From another aspect, percentage, we can gain a direct visual experience to see the model performance, while RMSE requires the experience for the scale and the dimension for the observations. In detail, model accuracy is considered excellent when RRMSE<10%; good in 10-20%; fair in 20-30% (Heinemann et al. 2012).

[1] https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit

$$RRMSE = \frac{RMSE}{\bar{O}} \times 100$$

Python code

```
# RRMSE (relative root mean square error)
elif type_statistic == '2':
    out = np.sqrt((diff**2).mean())
    out = out/mean_true*100
```

3) Mean absolute error (MAE)

Basic concepts

Compared with RMSE, mean absolute error (MAE) would be a better metric to indicate the average model performance, and the RMSE is by definition never smaller than the MAE (Chai, Draxler 2014). In (Chai, Draxler 2014), it is pointed out that MAE gives the same weight to all errors, while the RMSE gives more weight for the bigger errors. From the equations, we can also gain this point. The unit of MAE is the same as the observation. MAE is less sensitive to the effect of outliers than RMSE as an indicator of model performance (White et al. 2018).

Considering our project, such as leaf area index estimation via machine learning, outliers occur. Therefore, the RMSE is better.

$$MAE = \frac{\sum_{i=1}^{n}|O_i - E_i|}{n}$$

Python code

```
# MAE (mean absolute error)
elif type_statistic == '3':
    out = (abs(diff)).mean()
```

4) Correlation coefficient (r)

Basic concepts

From the book, Statistics for Environmental Engineers (Brown and Hambley 2002): 1) a statistic that quantifies the strength of the relationship between the variables, which showed a linear relationship, is the correlation coefficient; 2) Correlation here may, but does not necessarily, indicate causation, and this will be explained later; 3) A scaleless covariance, called the correlation coefficient $\rho(x, y)$ or simple ρ, is obtained by dividing the covariance by the two population standard deviations σ_x and σ_y, respectively. 4) The possible values of ρ range from -1 to +1. If x were independent of y, ρ would be zero. Value approaching -1 or +1 indicate a strong correspondence of x with y. a positive correlation $(0 < \rho \leq 1)$ indicates that the large values of x are associate with large values of y. in contrast, a negative correlation $(-1 \leq \rho < 0)$ indicates that the large values of x are associated with small values of y.

Two points from this book also need to gain our attention: 1) The correlation coefficient is a valid indicator of association between variables only when that association is linear. If it was not a linear relationship, the computed value of the correlation coefficient would not likely approach ±1, even if the experimental errors were vanishingly small; 2) Correlation, no matter how strong, does not imply causation. Regarding the relationship between correlation and causation, correlation is valid when both variables have random measurement errors. There is no need to think of one variable as X and the other as Y or of one as a predictor and the other as predicted. The two variables stand equal, and this helps remind us that correlation and causation are not equivalent concepts.

For our research, we have the equation below, and *r* is dimensionless.

$$r = \frac{cov(E,O)}{\sigma_E \sigma_O} = \frac{\sum_{i=1}^{n}(E_i - \bar{E})(O_i - \bar{O}) / n}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i - \bar{E})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - \bar{O})^2}} = \frac{\sum_{i=1}^{n}(E_i - \bar{E})(O_i - \bar{O})}{\sqrt{\sum_{i=1}^{n}(E_i - \bar{E})^2}\sqrt{\sum_{i=1}^{n}(O_i - \bar{O})^2}} \times$$

Python code

```
# r (correlation coefficient)
elif type_statistic == '4':
   tmp = np.corrcoef(true,pred)
   out = tmp[0,1]
```

5) Coefficient of determination (R^2)

Basic concepts

The equation below (Legates and McCabe 1999) shows how to calculate the coefficient of determination, and it is dimensionless.

$$R^2 = r^2 = \left\{ \frac{\sum_{i=1}^{n}(E_i - \bar{E})(O_i - \bar{O})}{\left[\sum_{i=1}^{n}(O_i - \bar{O})^2\right]^{0.5} \left[\sum_{i=1}^{n}(E_i - \bar{E})^2\right]^{0.5}} \right\}^2$$

The coefficient of determination is the square of Pearson's product-moment correlation coefficient (R^2=r^2), and it describes the proportion of the total variance in the observed data that can be explained by the model. It ranges from 0.0 to 1.0, with higher values indicating better agreement (Legates and McCabe 1999).

In (Brown and Hambley 2002): 1) the coefficient of determination is that proportion of the total variability in the dependent variable that is accounted for by the regression equation; 2) A value of R^2=1 indicates that the fitted equation accounts for all the variability of the values of the dependent variables in the sample data. At the other extreme, R^2=0 indicates that the regression equation explains none of the variability. This idea is so simple that we naturally tend to assume that a high R^2 assures a statistically significant regression equation and that a low R^2 proves the opposite; 3) A "statistically significant equation" would mean that we conclude there is some true relationship between the independent and dependent variables and that this relationship could be used to predict new conditions; 4) A high R^2 does not assure a valid relation; 5) A low R^2 does not mean the model is useless; 6) A significant R^2 does not mean the model is useful; 7) The magnitude of R^2 depends on the range of variation in X.

Python code

```
# R2 (coefficient of determination)
elif type_statistic == '5':
   tmp = np.corrcoef(true,pred)
   tmp = tmp[0,1]
   out = tmp**2
```

6) Coefficient of Efficiency (E)

Basic concepts

The coefficient of efficiency, E, has been widely used to evaluate the performance of hydrologic models (Legates and McCabe 1999). The range of the coefficient of efficiency starts from minus infinity to 1.0, with higher values indicating better agreement. The equation to calculate the coefficient of efficiency is below, and the coefficient of efficiency is dimensionless.

$$E = 1.0 - \frac{\sum_{i=1}^{n}(O_i - E_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$$

Python code

```
# E (coefficient of efficiency)
elif type_statistic == '6':
    out = 1 - (diff**2).sum()/((true - mean_true)**2).sum()
```

7) Mean squared error (MSE)

Basic concepts

MSE is similar to the RMSE, which gives more weight to larger differences. The smaller the means squared error, the closer you are to finding the line of best fit[2].

The mean squared error tells you how close a regression model is to a set of points. It does this by taking the distances from the points to the regression model (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. The unit is the square of the unit of the observation.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(O_i - E_i)^2$$

Python code

```
# MSE (mean squared error)
elif type_statistic == '7': # mean squarred error
    out = diff**2
    out = out.mean()
```

4. Example

The data used is “OTO.xlsx”, and the function is the same as it shown in Figure 1. Figure 2 shows how the observation vs. estimation looks like, and Figure 3 tells the goodness-of-fit statistics for this situation.

[2] https://www.statisticshowto.com/mean-squared-error/

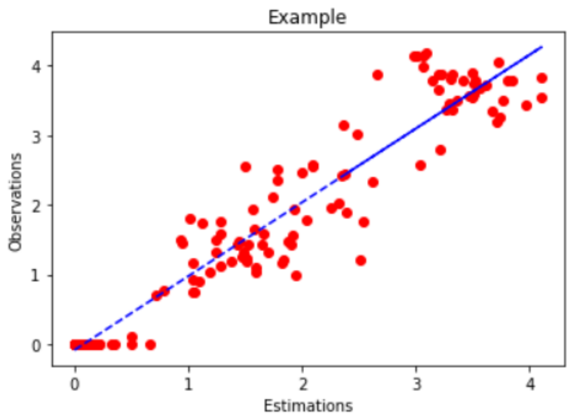
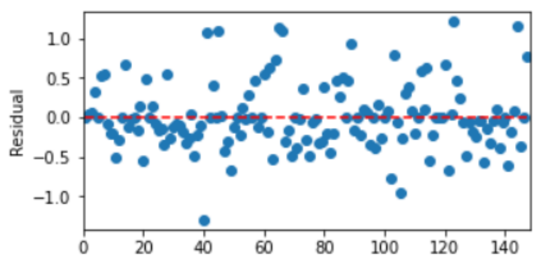


Figure 2. Observations vs. estimations




```
RMSE: 0.4191 RRMSE: 26.88 % MAE: 0.2988 r: 0.9596
R2: 0.9209 E: 0.9178 MSE: 0.1757
```

Figure 3. Residual plot and the goodness-of-fit statistics

5. Reference

[1] Aboutalebi, Mahyar et al. 2019. "Incorporation of Unmanned Aerial Vehicle (UAV) Point Cloud Products into Remote Sensing Evapotranspiration Models." *Remote Sensing* 12(1): 50. https://www.mdpi.com/2072-4292/12/1/50 (February 8, 2020).

[2] Brown, P. M. B. L. C., and D. F. Hambley. 2002. "Statistics for Environmental Engineers." *Environmental & Engineering Geoscience* 8(3): 244–45. http://pubs.geoscienceworld.org/aeg/eeg/article-pdf/8/3/244/3106832/i1078-7275-008-03-0244.pdf (November 4, 2020).

[3] T Chai, and R R Draxler. 2014. "Ozone Health and Ecosystem Impacts View Project Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?-Arguments against Avoiding RMSE in the Literature." *Geosci. Model Dev* 7: 1247–50. www.geosci-model-dev.net/7/1247/2014/ (November 3, 2020).

[4] Heinemann, Alexandre Bryan, Pepijn A.J. Van Oort, Diogo Simões Fernandes, and Aline de Holanda Nunes Maia. 2012. "Análise de Sensibilidade Do Modelo APSIM/ORYZA Na Estimava de Erros Na Radiação Solar." *Bragantia* 71(4): 572–82. www.simego.sectec.go.gov.br/ (November 4, 2020).

[5] Käfer, Pâmela Suélen, and Nájila Souza da Rocha. 2020. "Artificial Neural Networks Model Based on Remote Sensing to Retrieve Evapotranspiration over the Brazilian Pampa." *Journal of Applied Remote Sensing* 14(03): 038504. https://www.spiedigitallibrary.org/journals/journal-of-applied-remote-sensing/volume-14/issue-03/038504/Artificial-neural-networks-model-based-on-remote-sensing-to-retrieve/10.1117/1.JRS.14.038504.full (September 20, 2020).

[6] Legates, David R., and Gregory J. McCabe. 1999. "Evaluating the Use of 'Goodness-of-Fit' Measures in Hydrologic and Hydroclimatic Model Validation." *Water Resources Research* 35(1): 233–41. http://doi.wiley.com/10.1029/1998WR900018 (November 3, 2020).

[7] McKeen, S. et al. 2005. "Assessment of an Ensemble of Seven Real-Time Ozone Forecasts over Eastern North America during the Summer of 2004." *Journal of Geophysical Research Atmospheres* 110(21): 1–16. https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2005JD005858 (November 3, 2020).

White, William A. et al. 2018. "Determining a Robust Indirect Measurement of Leaf Area Index in California Vineyards for Validating Remote Sensing-Based Retrievals." *Irrigation Science* 37(3): 269–80.