A Forecasting Evaluation Library for *J*Demetra+

David de Antonio Liedo[*]

January 26, 2017

Abstract

This note presents the main tools currently available in *J*Demetra+ for the evaluation of forecasting errors. We explain the main statistical concepts and their implementation.

Key words: JDemetra+Nowcasting, forecasting evaluation, dynamic factor models, Diebold-Mariano Test, fixed-smoothing asymptotics.

Contents

[*] National Bank of Belgium, R&D Statistics, e-mail: david.deantonioliedo@nbb.be.

Evaluating Forecasting Accuracy: Concepts

The prediction errors are defined with a reference i to the information set available at the time the forecast was made: $e_{t|i} = y_t - \hat{y}_{t|\mathcal{F}_i}$, where $\mathcal{F}_i$ may include lags of y_t and many other variables that do not necessarily refer to the same period. In practice, the information that will be actually used may be a small subset of $\mathcal{F}_i$.

The properties of these forecast errors can be assessed in isolation or relative to a benchmark, which we will define as $\breve{e}_{t|i} = y_t - \breve{y}_{t|\mathcal{F}_i}$. The benchmark may be a naive forecast, e.g. random walk, in which case $\breve{y}_{t|\mathcal{F}_i}$ would be equal to $\breve{y}_{t|y_{t-1}} = y_{t-1}$. However, the benchmark could also be a prediction that is regularly published by a forecasting institute or market analysts, which is not necessarily model-based. In that case, $\breve{y}_{t|\mathcal{F}_i}$ would be given by methods and a subset of $\mathcal{F}_i$ which is unknown to us.

For model-based forecasts, we use the following notation: $\hat{y}_{t|\mathcal{F}_i} = E_\theta[y_t|\mathcal{F}_i]$ to highlight the fact that they are based on model-consistent expectations given by the parameter vector θ.

In forecasting comparisons involving competing forecasts that result from the same information set, the subindex i will be removed because it does not play a role. We will first test the following *null* hypotheses involving forecast errors:

$$Unbiasedness: \quad E[e_t] = 0 \tag{1}$$

$$Autocorrelation: \quad E[e_t e_{t-1}] = 0 \tag{2}$$

$$Equality\ in\ squared\ errors: \quad E[e_t^2 - \breve{e}_t^2] = 0 \tag{3}$$

$$Equality\ in\ absolute\ errors: \quad E[|e_t| - |\breve{e}_t|] = 0 \tag{4}$$

$$Forecast\ \hat{y}_t\ encompasses\ \breve{y}_t: \quad E[(e_t - \breve{e}_t)e_t] = 0 \tag{5}$$

$$Forecast\ \breve{y}_t\ encompasses\ \hat{y}_t: \quad E[(\breve{e}_t - e_t)\breve{e}_t] = 0 \tag{6}$$

The *null* hypothesis may be rejected in favour of either one or two-sided *alternatives*. For example, the hypothesis of equality in forecast errors originally proposed by Diebold and Mariano (see below) was tested against the two-sided alternative[1] (i.e. $\neq 0$).

An overview of the tests can also be found in Table A.1.

[1] In turn, one may decide to define a rejection only when the loss differential $e_t^2 - \breve{e}_t^2$ is negative, i.e. our squared forecast errors are smaller than those coming from the benchmark.

Diebold-Mariano Test

The test originally proposed by Diebold and Mariano (1995) considers a sample path of loss differentials $\{d_t\}_{t=1}^{T}$. In the case of a squared loss function, we have $d_t = e_t^2 - \breve{e}_t^2$. Under the assumption that the loss differential is a covariance stationary series, the sample average, $\bar{d}$, converges asymptotically to a normal distribution:

$$\sqrt{T}\bar{d} \ \underset{\rightarrow}{d} \ N(\mu, 2\pi f_d(0)) \tag{7}$$

In particular, they proposed to test the null hypothesis that the forecast errors coming from the two forecasts bring about the same loss: $E[e_t^2 - \breve{e}_t^2] = 0$ against the two-sided alternative. Thus, the resulting p-values represent the probability of obtaining the realized forecast error differential or a more extreme one in a new experiment if the null hypothesis was actually true. The test-statistic that will be used to calculate our p-values is computed as follows:

$$DM = \frac{\bar{d}}{\sqrt{\dfrac{2\pi\hat{f}_d(0)}{T}}} \tag{8}$$

where $2\pi\hat{f}_d(0)$ is a consistent estimate of the variance of $\bar{d}$. Consider $2\pi\hat{f}_d(0) = \sum_{\tau=-(T-1)}^{(T-1)} w_\tau \gamma_d(\tau)$, where $\gamma_d(\tau) = \dfrac{1}{T}\sum_{t=|\tau|+1}^{T}(d_t - \bar{d})(d_{t-|\tau|} - \bar{d})$. Under the assumption that $\gamma_d(\tau) = 0$ for $\tau \geq h$, we can use a rectangular lag window estimator by setting $w_\tau = 0$ for $\tau \geq h$. Another option is to use the Heteroscedasticity and Autocorrelation Consistent (HAC) estimator proposed by Newey and West (1987). In this case, the weights could be given by a triangular window, $w_\tau = 1 - \dfrac{\tau}{h}$ for $\tau < h$. In this case, however, the consistency property only remains valid when the truncation lag h or bandwidth is a function of the sample size T.

The idea is to test the statistical significance of the regression of $e_t^2 - \breve{e}_t^2$ on an intercept. In order to determine the statistical significance of the intercept, its associated standard errors need to take into account the autocorrelation patterns of the regression error, which are considered in the denominator of equation (8). JDemetra+ exploits the same unified framework to conduct all tests listed in Table A.1. But given the small sample sizes that are typical in real-time forecasting applications, which leads to an over-rejection of the

Table A.1: Forecasting Evaluation Tests

Test	Null hypothesis	Statistic	Asym. theory	Finite sample	JDemetra+ implementation
Bias	$E[e_t] = 0$	$\mathrm{B} = \dfrac{\bar{e}}{\sqrt{\dfrac{2\pi \hat{f}_e(0)}{T}}}$	$N(0,1)$	KV(2005)	`BiasTest`
Autocorrelation	$E[e_t e_{t-1}] = 0$	$\mathrm{AR} = \dfrac{\bar{\rho}}{\sqrt{\dfrac{2\pi \hat{f}_\rho(0)}{T}}}$	$N(0,1)$	KV(2005)	`EfficiencyTest`
Diebold-Mariano	$d_t \equiv L_{1,t} - L_{2,t} = 0$	$\mathrm{DM} = \dfrac{\bar{d}}{\sqrt{\dfrac{2\pi \hat{f}_d(0)}{T}}}$	$N(0,1)$	KV(2005)	`DieboldMarianoTest`
Encompassing 1	$d^e_{1,t} \equiv E[(e_t - \breve{e}_t)e_t] = 0$	$\mathrm{E}_1 = \dfrac{\bar{d}_1}{\sqrt{\dfrac{2\pi \hat{f}_{d_1}(0)}{T}}}$	$N(0,1)$	KV(2005)	`EncompassingTest`
Encompassing 2	$d^e_{2,t} \equiv E[(\breve{e}_t - e_t)\breve{e}_t] = 0$	$\mathrm{E}_2 = \dfrac{\bar{d}_2}{\sqrt{\dfrac{2\pi \hat{f}_{d_2}(0)}{T}}}$	$N(0,1)$	KV(2005)	`EncompassingTest`

null hypothesis, we follow the fixed-smoothing (FS) asympotics proposed by Coroneo and Iacone (2015) exploiting the finite sample distributions of Kiefer and Vogelsang (2005). The distribution of the test statistic (8) will depend on kernel (triangular in our case) and the bandwidth chosen, which is set by default equal to $T^{0.5}$, as suggested by Coroneo and Iacone (2015). The results can be very different than those resulting from the traditional asymptotic theory, where the test statistic would have the same distribution under the null independently of the kernel and the bandwidth used.

Encompassing Test

Independently of whether the null hypothesis $E[e_t^2 - \breve{e}_t^2] = 0$ is rejected or not, it is relevant to understand to what extent our model encompasses all the relevant information of the benchmark, and the other way around. Because of the obvious symmetry of both statements, we consider only the first one. If our forecast $y_{t|\mathcal{F}_i}$ encompasses a given benchmark $\breve{y}_{t|\mathcal{F}_i}$, the difference between those benchmark forecasts and ours will not be a relevant factor in explaining our own forecast error. In other words, the regression coefficient λ will not be significantly different from zero in the following regression:

$$\underbrace{y_t - y_{t|\mathcal{F}_i}}_{e_t} = \lambda \underbrace{(\breve{y}_{t|\mathcal{F}_i} - y_{t|\mathcal{F}_i})}_{e_t - \breve{e}_t} + \xi_t \tag{9}$$

$$\Updownarrow$$

$$y_t = \lambda \breve{y}_{t|\mathcal{F}_i} + (1-\lambda) y_{t|\mathcal{F}_i} + \xi_t \tag{10}$$

Following Harvey, Leybourne and Newbold (1997), the statistical significance of the λ coefficient in regression 9 can be used to reject the null hypothesis that our model encompasses the benchmark. In this case of rejection, equation 10 suggests that a combination of the two forecasts would yield a more informative forecast.

By construction, the value of the coefficient of a regression $\breve{e}_t = \alpha(\breve{e}_t - e_t) + \xi_t$ is equal to $1 - \lambda$, but it is not necessarily true that the *rejection* of the null hypothesis in the first case implies the *acceptance* of the symmetric statement.

The test-statistic is computed as follows. When the null hypothesis is that our model encompasses the benchmark, we define the sequence $\{d_t\}_{t=1}^T$, where $d_t = e_t(e_t - \breve{e}_t)$, and we compute $E1 = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{T}}}$, exactly as in equation 8.

Efficiency: Bias Test

In order to assess whether our forecasts are unbiased, we will simply test the statistical significance of the average error. In some cases, the time series of forecast errors $\{e_t\}_{t=1}^{T}$ may be autocorrelated to some extent even when they are based on a model with IID innovations. In such cases, the variance associated to the estimate of the average forecast error may be large. The test statistic has exactly the same form as the previous tests discussed so far.

Efficiency: Autocorrelation Test

We will test here a second necessary condition for our forecasts to be efficient: absence of autocorrelation. In the same spirit as the tests described above, we will assess the statistical significance of the forecast errors' autocorrelation. Thus, our sequence $\{d_t\}_{t=1}^{T}$ will be defined with $d_t = e_t e_{t-1}$.

Implementation in *J*Demetra+

Structure of the library

The code is structured as follows:

1. The class `AccuracyTests` contains all functions required to compute the statistic defined in equation (8). It also incorporates instructions regarding how to calculate the pvalues when depending on whether one wants to use standard asymptotics or fixed-smoothing asymptotics (the input `AsymptoticsType` is required).

2. The class `AccuracyTests` is extended by each one of the classes containing the tests listed in Table (A.1). The constructor of each one of these classes can generate the tests when either the forecasts or the forecast errors are given as an input. These classes also define the method to calculate the loss function d_t, which is specific to each test.

3. The class `GlobalForecastingEvaluation` can also be used to generate all the tests. It contains `BiasTest, EfficiencyTest, DieboldMarianoTest` and `EncompassingTest` and the corresponding methods to get the test statistics and p-values described in Table A.1.

4. The class `ForecastEvaluation` contains methods to quantify errors: Root Mean Squared Errors (RMSE), relative RMSE, Mean Absolute Errors (MAE), etc...

A simple example

Suppose we want evaluate the forecast of a model (f_t^m) and compare them with those of a benchmark (f_t^b). The following points explain all the steps followed in the code below to run all the tests:

- First we need to initialize the two competing forecast (i.e. benchmark vs model), all the statistics we are going to calculate (RMSE, bias, autocorrelation, and encompassing weights) and the p-values corresponding to each one of the tests.

- Second, we initialize the `eval` object of the class `GlobalForecastingEvaluation`, which will contain all test results. The inputs needed to run the tests are three time series (our model's forecasts, those of the benchmark, and the actual data, which is the target) and the kind of distribution of the various test statistics under the null, which is given by a normal distribution when

  ```
  AccuracyTests.AsymptoticsType.STANDARD_FIXED_B
  ```

  is used.

- By choosing the option

  ```
  AccuracyTests.AsymptoticsType.HAR_FIXED_B
  ```

  the distribution tabulated by Kiefer and Vogelsang (2005) is used.

- For each type of test, the bandwidth used to estimate the variance needs to be specified. Otherwise, the default value will be used ($T^{1/2}$). The relevant statistics for each test as well as the pvalues are obtained with a simple get command. Notice that getPValue(twoSided) uses the logical argument true in order to get the p-values of the two-sided test.

```
public void example() {

TsData[] series = {benchmark, model, target};

boolean twoSided = true;

double rmse = new double ;
double dmPval = new double ;

double bias = new double ;
double biasPval = new double ;

double arcorr = new double ;
double arPval = new double ;

double m_enc_bench = new double ;
double m_enc_bench_Pval = new double ;
double bench_enc_m = new double ;
double bench_enc_m_Pval = new double ;

// squared root of T
int bandwith = (int) Math.pow(series.getObsCount(), 1.0 / 2.0);

GlobalForecastingEvaluation eval = new GlobalForecastingEvaluation(model, benchmark,
    target,
AccuracyTests.AsymptoticsType.HAR_FIXED_B);
eval.getDieboldMarianoTest().setBandwith(bandwith);

dmPval = eval.getDieboldMarianoTest().getPValue(twoSided);
ForecastEvaluation feval = new ForecastEvaluation(model, benchmark, target);
rmse = feval.calcRMSE();

eval.getBiasTest().setBandwith(bandwith);
bias = eval.getBiasTest().getAverageLoss();
biasPval = eval.getBiasTest().getPValue(twoSided);

eval.getEfficiencyTest().setBandwith(bandwith);
arcorr = eval.getEfficiencyTest().calcCorelation();
arPval = eval.getEfficiencyTest().getPValue(twoSided);

eval.getModelEncompassesBenchmarkTest().setBandwith(bandwith);
m_enc_bench = eval.getModelEncompassesBenchmarkTest().calcWeights();
m_enc_bench_Pval = eval.getModelEncompassesBenchmarkTest().getPValue(twoSided);
bench_enc_m = eval.getBenchmarkEncompassesModelTest().calcWeights();
bench_enc_m_Pval = eval.getBenchmarkEncompassesModelTest().getPValue(twoSided);
}
```

Example in the context of nowcasting

Let's use a more complex example taken from Basselier, de Antonio and Langenus (2017) in the context of a model for nowcasting, where blocks of data releases are used to update the predictions. The aim is to determine which blocks of releases lead to significant improvements of the forecasting accuracy.

The code has been reproduced below. The first variable `series` is an array of time series, which includes the forecasts of an ARIMA model followed by the nowcasts from a dynamic factor model obtained at different points in time, ranging from 90 days before the end of the quarter (DFMPRE90) to 44 days after the end of the quarter (DFM$POST$44). The variable FLASH corresponds to the GDP growth, which is the target, or the variable the model aims to predict.

The following code uses a simple loop to produce the test results for the eleven DFM updates. For each one, we not only test the bias and autocorrelation, but we also check whether it adds any value with respect to the previous forecast by means of both the Diebold-Mariano and Encompassing tests. For example, is DFMPRE90 better than ARIMA? Is DFMPRE75 better than DFMPRE90? More precisely, as explained above, the Diebold-Mariano test simply evaluates whether the difference in accuracy for each pair is statistically significant, while the encompassing test checks whether the updated forecast can be explained by its deviation with respect to the old one. Rejecting this hypothesis implies that the new forecast encompasses the old one (i.e. the weight λ in equation 10 above is not significantly different from zero).

```
public void testingUpdates() {

TsData[] series = {ARIMA, DFMPRE90, DFMPRE75, DFMPRE60, DFMPRE45, DFMPRE30, DFMPRE15,
    DFM0, DFMPOST15, DFMPOST30, DFMPOST42, DFMPOST44, FLASH};

boolean twoSided = true;
double[] rmse = new double[11];
double[] dmPval = new double[11];

double[] bias = new double[11];
double[] biasPval = new double[11];

double[] arcorr = new double[11];
double[] arPval = new double[11];

double[] m_enc_bloom = new double[11];
double[] m_enc_bloom_Pval = new double[11];
double[] bloom_enc_m = new double[11];
double[] bloom_enc_m_Pval = new double[11];

for (int i = 1; i < series.length; i++) {
// squared root of T
int bandwith = (int) Math.pow(series[i].getObsCount(), 1.0 / 2.0);

GlobalForecastingEvaluation eval = new GlobalForecastingEvaluation(series[i], series[i -
    1], FLASH, AccuracyTests.AsymptoticsType.HAR_FIXED_B);

eval.getDieboldMarianoTest().setBandwith(bandwith);
dmPval[i] = eval.getDieboldMarianoTest().getPValue(twoSided);
ForecastEvaluation feval = new ForecastEvaluation(series[i], ARIMA, FLASH);
rmse[i] = feval.calcRMSE();

eval.getBiasTest().setBandwith(bandwith);
bias[i] = eval.getBiasTest().getAverageLoss();
biasPval[i] = eval.getBiasTest().getPValue(twoSided);

eval.getEfficiencyTest().setBandwith(bandwith);
arcorr[i] = eval.getEfficiencyTest().calcCorelation();
arPval[i] = eval.getEfficiencyTest().getPValue(twoSided);

eval.getModelEncompassesBenchmarkTest().setBandwith(bandwith);
m_enc_bloom[i] = eval.getModelEncompassesBenchmarkTest().calcWeights();
m_enc_bloom_Pval[i] = eval.getModelEncompassesBenchmarkTest().getPValue(twoSided);
bloom_enc_m[i] = eval.getBenchmarkEncompassesModelTest().calcWeights();
bloom_enc_m_Pval[i] = eval.getBenchmarkEncompassesModelTest().getPValue(twoSided);

System.out.println("RMSE" + "\t" + rmse[i] + "\t" + dmPval[i] + "\t" + "Bias" + "\t" +
    bias[i] + "\t" + biasPval[i] + "\t" + "CORR" + "\t" + arcorr[i] + "\t" + arPval[i] +
    "\t" + "Weight on Update" + "\t" + (1 - m_enc_bloom[i]) + "\t" + "M_enc_bench Pval" +
    "\t" + "Bench_enc_M Pval" + "\t" + bloom_enc_m_Pval[i]);
}
}
```

The outcome of theses tests has been organized in Table A.2. It includes bias, autocorrelation, RMSE and the λ coefficient defined above, which is the weight given to a benchmark forecasts that competes with their model's. Statistical significance is highlighted with shades. Grey shaded areas in column FS-DM demonstrate which news blocks have induced a significant change in the RMSE of the model, i.e. the null of equal accuracy between old (O) and updated (U) forecasts is rejected. The outcome of the DM test may be considered jointly with the results of the encompassing tests. For a certain news block to be considered relevant, the corresponding nowcasting update (U) should hold a larger amount of information than the older nowcast (O) based on the previous information set, while the old nowcast does not incorporate any useful information absent in the new update. The last two colums of the table show that this is generally the case, with some exceptions. That is, the null *U encompases O* is not rejected while *O encompasses U* is rejected.

Table A.2: Statistical significance of each update based on fixed-smoothing (FS) asymptotics

Evaluation period: 2007.Q1 - 2015.Q1, T=25

	FS-Efficiency		**FS-DM**	**FS-Encompassing (U)pdate vs (O)ld**	
Real-Time Updates	bias	corr	RMSE	U enc O	O enc U
ARIMA	-0.27	0.50	-	-	-
DFM -90 (d)ays	-0.22	0.41	0.68	0.60	0.39
DFM -75 d	-0.19	0.47	0.55	-0.60	1.59
DFM -60 d	-0.12	0.55	0.52	0.26	0.51
DFM -45 d	-0.14	0.54	0.54	1.48	-0.54
DFM -30 d	-0.08	0.58	0.50	-0.20	1.07
DFM -15 d	-0.13	0.46	0.41	-0.65	1.59
DFM 0 d (end of quarter)	-0.06	0.45	0.38	-0.13	0.82
DFM +15 d	-0.09	-0.11	0.27	-0.02	1.01
DFM +30 d	-0.07	-0.08	0.26	-0.39	1.23
DFM +42 d	-0.10	-0.06	0.26	0.27	0.66
DFM +44 d	-0.06	-0.18	0.23	-0.17	1.03

Note: The FS-Efficiency multicolumn of his table reports bias and autocorrelation for the forecast errors obtained at different horizons. The FS-DM and FS-Encompassing blocks should be considered simultaneously. They aim to determine for each forecasting update (U) whether there is any added value with respect to the old/last available forecast (O). The null hypothesis of the Diebold-Mariano (DM) test is rejected when the *difference in the squared errors of U and O* is significantly different from zero. For the two encompassing tests, the null hypothesis states that the updated forecast (U) encompasses all the relevant information from the old forecast (O) (*or vice versa*). When the null hypothesis can be rejected, this implies that *U can be improved by combining it with O*. The combination weight associated to O (*or U*) is therefore reported below the "U enc O" test. In order to assess the added value of the updated forecast, the DM null of equal forecast accuracy should be rejected and at the same time the null "U enc O" and "O enc U" should be, respectively, not rejected and rejected. Given the small size of our evaluation sample and the time-series correlation patterns, we determine significance at the 5%, 10% and 20% level using the fixed-smoothing (FS) asymptotics, as proposed by Coroneo and Iacone (2015).

References

[1] Coroneo, L. and F. Iacone (2016). “Comparing predictive accuracy in small samples using fixed-smoothing asymptotics”. *Discussion Papers University of York*, **15/15**.

[2] Basselier R., D. de Antonio Liedo, and G. Langenus (2015). “Nowcasting Euro Area GDP: Understanding the Role of Qualitative Surveys”. , , x-X

[3] Diebold, F.X. and R.S. Mariano (1995). “Comparing Predictive Accuracy”. *Journal of Business and Economic Statistics*, **13**, 253-265.

[4] Kiefer, N.M., and T.J. Vogelsang (2005). “ A new asymptotic theory for heteroskedasticity-autocorrelation robust tests”. *Econometric Theory*, **21**, 1130-1164.

[5] Newey, W.K. and K.D. West (1987). “A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix”. *Econometrica*, **55(3)**, 703-708.