Problem Set 3: Forecasting

Due February 27

Assignment:

1. Undergraduates: Do all four problems.
2. Math graduate students: Do all four problems.

You are encouraged to work other students but turn in your own paper. Please—no electronic copies. This problem set asks you to work with the Python script `ibm.py` and the data file `ibm.txt`, both posted on the Moodle webpage.

1. A very useful measure of forecasting accuracy is the coefficient of determination. It compares the sum of squared errors produced by the prediction/forecasting function to the sum of squared errors produced by a baseline prediction function. Often, the baseline prediction function is the sample mean (hence, the sample mean of the target sequence is set to be the prediction for every instance). The coefficient of determination can be computed as follows:

   $$r^2 = \frac{\sum_i (y_i - \overline{y})^2 - \sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}. \tag{1}$$

   Thus, we see that it compares two predictive functions: the forecasting function in question yielding the forecasts $\widehat{y}_1, \ldots, \widehat{y}_n$ to the sample mean, yielding the prediction $\overline{y}$ for every target. The sample mean may be a terrible forecasting function *especially* when the process generating the target values is subject to trend or drift.

   A more reasonable and simple baseline prediction function is an exponentially weighted forecasting function. The exponentially weighted forecast at time step $n+1$ is a weighted average of all target values observed up to and including the most recent observation (the nth observation). Notationally, the exponentially weighted forecast is

   $$\widehat{y}_{n+1} = \widehat{\mu}_n = \sum_i^n w_i y_i, \tag{2}$$

   where the ith weight is $w_i = \alpha(1-\alpha)^{n-i}$. Consequently, $\widehat{\mu}_n$ is a time-varying estimate of the process mean at time step n. An alternative expression is

   $$\widehat{\mu}_n = \alpha y_n + (1-\alpha)\widehat{\mu}_{n-1}. \tag{3}$$

   The tuning constant α is a real number between 0 and 1. The choice of α determines the smoothness of the forecasting function. Larger values of α place greater weight on more recent observations and imply that the process mean is changing rapidly with time step.

The exponentially weighted forecast at any time step following n is the same: $\widehat{\mu}_n$. If the process is drifting, then this forecast is not good for time steps removed from the time step of the last observation. There's no opportunity to account for trend in the series. This deficiency leads to the Holt-Winters forecasting function. It includes a time-varying slope $\widehat{\beta}_n$. Two tuning constants are needed: $0 < \alpha_h < 1$ and $0 < \alpha_b < 1$. The forecast for τ time steps ahead is

$$\widehat{y}^H_{n+\tau} = \widehat{\mu}_n + \tau\widehat{\beta}_n, \tag{4}$$

where

$$\widehat{\mu}_n = \alpha_h y_n + (1 - \alpha_h)(\widehat{\mu}_{n-1} + \widehat{\beta}_{n-1}), \tag{5}$$

and

$$\widehat{\beta}_n = \alpha_b(\widehat{\mu}_n - \widehat{\mu}_{n-1}) + (1 - \alpha_b)\widehat{\beta}_{n-1}. \tag{6}$$

The term $\widehat{\mu}_n - \widehat{\mu}_{n-1}$ is the most recent estimate of trend whereas $\widehat{\beta}_{n-1}$ is the trend estimate determined by the past observations (with greater weight assigned to the more recent observations.) Note that only the target sequence is used (as if there were no useful predictor variables) as is the case with $\overline{y}$, the first baseline forecasting function.

Write Python code to carry out exponential and Holt-Winters forecasting for the IBM data. Specifically,

(a) Compute the coefficient of determination for an exponential forecasting function, i.e., compute

$$r^2 = \frac{\sum_i (y_i - \overline{y})^2 - \sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{7}$$

where $\widehat{y}_i$ is obtained from one time step-ahead exponential forecasting. Find a good value for α, and report it and r^2.

(b) Repeat (a), but use Holt-Winters forecasting to compute $\widehat{y}_i$. Set $\alpha_h = \alpha/2$, where α was determined in part (a), and find a good value for α_b. Report α_h, α_b and r^2.

(c) Select a new baseline model to assess accuracy (in place of the $\overline{y}$ baseline model) from either your exponentially weighted forecasting function or your Holt-Winters forecasting function. Identify your forecasting function and report the tuning constants. Provide a plot of your baseline forecasting function and the targets. (You'll use it in the next two exercises).

2. Compare four forecasting functions using r^2: no interaction predictors (a total of 4 predictors: $\mathbf{x}_{\cdot 1}$, $\mathbf{x}_{\cdot 2}$, $\mathbf{x}_{\cdot 3}$, and $\mathbf{x}_{i,4}$), one interaction predictor and the terms from which it was constructed, say, $\mathbf{x}_{\cdot,1\cdot 2}$, $\mathbf{x}_{\cdot 1}$, and $\mathbf{x}_{\cdot 2}$, another one-interaction forecasting function built from three terms, for instance $\mathbf{x}_{\cdot 3}$, $\mathbf{x}_{\cdot 4}$, and $\mathbf{x}_{\cdot,3\cdot 4}$, and lastly, a function utilizing all six predictor variables from the two interaction functions (for instance, $\mathbf{x}_{\cdot,1\cdot 2}$, $\mathbf{x}_{\cdot 1}$, $\mathbf{x}_{\cdot 2}$, $\mathbf{x}_{\cdot 3}$, $\mathbf{x}_{\cdot 4}$, and $\mathbf{x}_{\cdot,3\cdot 4}$). Create and report a table listing the predictor variables for each forecasting function and the associated r^2 values where r^2 is computed using the baseline model from 1.(c) above.

3. Use one of the forecasting functions from the previous exercise and introduce a regularization term into the objective function (refer to Problem Set 2, #3). By trial and error, see if you can improve the forecasting function by your choice of λ. Report r^2 as computed from the baseline model given in 1.(c) above and your best λ. Provide a plot of the target sequence and the regularized forecasts.

4. Another approach is to construct variables from the target sequence alone (hence, no predictor variables) that are capable of representing smooth, nonlinear trend in the sequence of target values. You are to so in this exercise by constructing a linear regression forecasting function built from cubic splines (the code is included in `ibm2.py` posted on the Moodle website).

   (a) Find a good linear regression forecasting function, describe the knots and the number of observations (`retain`) used to form the moving window (`xSub`). Report r^2 as computed from the baseline model given in 1.(c) above and provide a plot of the target sequence and the forecasts.

   (b) Include one or two predictor variables into the linear regression function (keeping the cubic spline terms) with the intent of improving the forecasting function. Can you get any improvement? Report r^2 and provide a plot of the target sequence and the cubic spline forecasts.