11/4/2016

**Residual Analysis**

Residual is the difference between the observed value of dependent variable and the predicted value. A residual plot is a graph that shows residual on the vertical axis and the independent variable on the horizontal axis.

Residual analysis typically is used in a linear regression model; we check residual plot to verify if linear relationship is appropriate for the model. If residuals are randomly dispersed around horizontal axis with mean of 0, linear is appropriate, otherwise non-linear is appropriate.

We can also understand residuals from randomness point of view. There are two basic components of valid regression model:

Response = (Constant + Predictors) + Error = Deterministic + Stochastic

Stochastic means random and unpredictable, residual should be stochastic, in other words, if you observe explanatory or predictive power in residual, you know that your predictors are missing some of the predictive information. The process is easy to understand with a die-rolling analogy, when you roll a die you shouldn’t be able to predict which number on any given toss, if the number six shows up more frequently than randomness dictates, you know sth wrong with your understanding of how the die actually rolls, if a gambler analyzes at the die rolls, he could adjust his mental model and play style to factor in the higher frequency of sixes, and his new mental model better reflects the outcome. The same principles apply to regression models, you shouldn’t be able to predict error for any given observation, just like the die, if the residuals suggest that your model is systematically incorrect, you have an opportunity to improve your model.

Non-random pattern of residuals indicates the predictor variables of the model did not capture some explanatory information that is leaking into the residuals, there are several possibilities to the non-randomness:

Missing variable

Missing higher-order term of variable in the model to explain curvature

Missing interaction between terms

In addition to above, there are two more ways that predictive information can sneak into the residuals:

The residuals should not be correlated with another variable

Adjacent residuals should not be correlated with each other(autocorrelation): if we can use one residual to predict the next residual, there is some specific information present that is not captured by the predictors, typically this situation involves time ordered observations. For example, if a residual is more likely to be followed by another residual that has the same sign, adjacent residuals are positively correlated, we can use a time variable to capture relevant time correlated information or use time series model.

**Heteroscedasticity** refers to the circumstance in which the variability of a variable is unequal across the range of values of a second variable that predicts it. The opposite of heteroscedasticity is **homoscedasticity**.

11/6/16

**Residual Analysis Tests**

<http://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm>

Test for Residual Normality, the two most common types are histogram, normal probability plot (QQ plot, an approximately straight line should be produced if the points come from a normal distribution), QQ plots work better than histogram on small sample size of residual.

Small departures from the straight line in the normal probability plot are common, but a clearly S curve suggests a bimodal distribution of residuals.

We can also test independence of residuals over time, if the order of observation represents the order of execution of each treatment combination.

We also plot residuals versus corresponding predicted values (residuals vs fit), this should produce a distribution of points scattered randomly about 0, commonly though residuals may increase as the value of fit increases. When this happens, it suggests one should transform the response, perhaps by modelling its logarithm or square root.

Another important test of residuals is to test it against predictor variables to assure no explanatory information sneaked into predictors from residuals.

**Autocorrelation in the residuals**(Time Series Residual Analysis)

<https://www.otexts.org/fpp/5/4>

When the data are a time series, you should look at ACF plot of the residuals. They will reveal if there is autocorrelation in the residuals, another test of autocorrelation is Durbin-waston test, it will report P value, if p is small, there is significant correlation remaining in the residuals.

**Sensitivity Analysis**

Sensitivity analysis is an analysis method that is used to identify how much variations in the input values for a given variable will impact the result. It is concerned with the uncertainty inherent in mathematical models where the values for inputs can vary, conclusion drawn from studies can be significantly altered depending on such things as how a certain variable is defined or observed or measured for a study, when results of study do not significantly change due to variations in underlying assumptions, they are considered robust.

11/10/2016

**Strategies selecting predictors**

<https://www.otexts.org/fpp/5/3>

**What are not recommended to select predictors:**

One common approach not recommended is to plot predictor against forecast variable and see if it shows noticeable relationship, this is invalid because it is not always possible to see the relationship from scatter plot, especially when the effect of other cofounding predictors have not been accounted for.

The other approach not valid is to build a multiple var regression model and remove predictors whose P value is above 0.05, this was because statistical significance does not always indicate predictive value, and P value can be misleading when confounding / collinearity are found between predictors. See <https://www.otexts.org/fpp/5/7/>

R square is not a good measure of predictive ability of a model either, imagine a model which produces forecasts that are exactly 20% of the actual values, in that case R square will be one, in addition, R square doesn’t account for degree of freedom, as adding any variable tends to increase R square even if that var is irrelevant.

People tends to use SSE sum of squared error based metric, but minimizing SSE is equivalent to maximizing R square and will always choose the model with most variables, and so is not a valid way selecting predictors. (if not using CV)

What are valid ways selecting predictors:

Adjusted R square punishes added predictor, maximizing adjusted R square is equivalent to minimizing the variance of SSE ( SSE/ ( N- k – 1), N is the number of observations, k is the number of predictors.

CV is a good way to choose predictors

Akaike’s information criterion, Corrected AIC, Bayesian Information Criterion are good ways.

AIC = Nlog(SSE/N) + 2(k+2)

AICc = AIC + 2(k+2)(k+3)/N-K-3;

BIC = Nlog(SSE/N) + log(N)(K+2)

AIC tends to select too many predictors when N is small, so a bias-corrected version AICc is developed. The model chosen by BIC either is the same with AIC or the one with smaller terms, as BIC tends to penalize the number of parameters more heavily, when N is large, BIC is similar to leave-v-out-CV when v = N(1-1/(logN-1)).

If N is large, AIC, AICc, BIC will lead to the same model. Adjusted R tends to select too many predictors compared to other methods, BIC tends to select too few predictors.

If K is large say we have 40 vars, it is not possible to fit all possible models as we would have 2^40 > 1 trillion possible models, we can use backwards stepwise regression: we start with model containing all vars, remove one at a time, keep iterating if it improves the measure of predictive accuracy, until no further improvement. If k is too large, this backwards step wise regression will not work, and we then start with only a subset of potential predictors, while backwards subtracting var one by one, an extra step needs to be inserted in which predictors are also added one at a time, with model being retained if it improves the measure of accuracy.

**Causation, Correlation, Multicollinearity**

<https://www.otexts.org/fpp/5/7/>

It is import not to confuse correlation with causation, or causation with forecasting. An var X maybe useful predicting a var Y, but that doesn’t mean X is causing Y, the relationship between X and Y could be more complicated than causality.

For example, it is possible to model the number of drownings at a beach resort each month with the number of ice creams sold in the same period, but that not because ice creams causing drownings, but because the temperature is high, people like to buy more ice creams and swim more. It is important to understand that correlation is useful in forecasting even when there is no causal relationship. However, a better model is possible if a causal mechanism can be determined.

We say two variables are confounded when their effects on the forecast variables cannot be separated, but we wouldn’t describe the confounded unless there was a relatively high level of correlation between them. Suppose we are forecasting monthly sales of a company for 2012, using data from 2000- 2011, in Jan 2009 a competitor came in the market taking some market share, and at the same time economy began to decline, it will not be possible to separate the effects of these two predictors because they are correlated.

A closely related issue is multicollinearity which occurs when similar information is provided by two or more predictors, it can occur in a number of ways: one is two predictors are highly correlated, one is a linear combination of predictors is highly correlated with another linear combination of predictors, in this case knowing the value of the first group of predictors tell you about the second group, hence they are providing similar information. The dummy variable trap is a special case of multicollinearity, suppose we have D1, D2, D3, D4, and D4 = 1 – D1 – D2- D3, so there is a perfect relation between D4 and D1+D2+D3.

Multicollinearity causing issues in the following ways:

If there is a perfect correlation (1/ -1), it is impossible to estimate regression model;

If there is a high correlation, estimating coefficients is computationally difficult;

The uncertainty associated with individual regression coefficients will be large due to the difficulty to estimate and thus forecasts will be unreliable. Forecasts will be unreliable especially when values are outside of the range of historical values of the predictors, it is always a bit dangerous when future values of the predictors lie much outside the historical range, but it is especially problematic when multicollinearity is present.

How does auto.arima work?

04/03/2017

Cuda is a parallel computing platform and programming model that makes using a GPU for general purpose computing simple and elegant. Cuda is not a language or an API.

GPU-accelerating computing is the use of GPU together with a CPU to accelerate scientific, engineering, mobile and enterprise applications, by offloading compute-intensive portions to applications of GPU, while the remainder of the codes still runs on the CPU.



**CPU VS GPU**

A CPU consists of a few cores optimized for sequential series processing while a GPU consists of thousands of smaller, more efficient cores designed for handling multiple tasks simultaneously.

**How to accelerate applications**

If GPU-acceleration is not ready available, you may be interested in developing GPU-accelerated code yourself, there are 3 main methods to achieve GPU-acceleration in your code.



Libraries:

Libraries give access to many different types of functions and algorithms that you do not have to implement in your software. Libraries are typically highly optimized and are accessed through API, making use of GPU acceleration libraries is typically the quickest way to add acceleration to your application. In fact, there are a number of GPU-acceleration libraries that are API compatible with CPU, this means you simply change the library you are compiling against- no codes change necessary, one example is OpenCV.

Compiler Directives:

Using compiler directives is a more flexible approach, we provide hints via complier directives( or pragmas) to tell the complier where and how it should parallelize compute intensive code for execution on an accelerator.

Compiler directives generally are associated with C,C++, Fortran, a similar approach in Python is use @vectorize decorator and the Continuum Numba compiler.

Programming:

Programming for the GPU in a CUDA-enabled language is the most flexible of the three approaches, we can write CUDA code in Python, and the code will be compiled using Numba compiler.

<https://www.youtube.com/watch?v=KM-zbhyz9f4>