

21: Validating Inferentials

In the previous article we covered the application of regression analysis to the development of inferential properties. Here we focus on their validation, prior to commissioning

DEVELOPERS of inferentials tend to demonstrate their reliability by plotting a line graph of the inferred and measured property. Figure 1 shows an example. It might appear that the inferential is reasonable at following the measured property, but it can be something of an illusion.

Figure 2 plots exactly the same data, but as a scatter chart. So, for example, if our inferential reports a value of 50%, the true value might be anywhere between 30 and 70%. However, this does not mean it is without value, remembering that the main role of the inferential is to give an early indication of a

Figure 1: Line graph

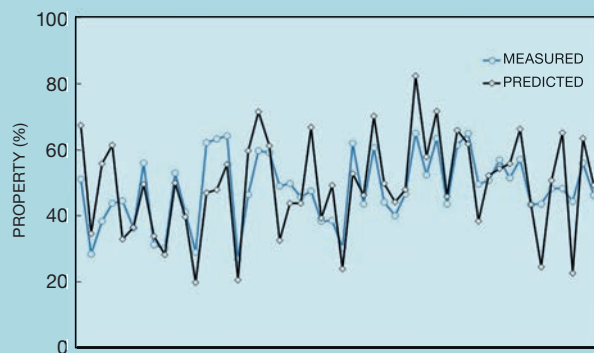
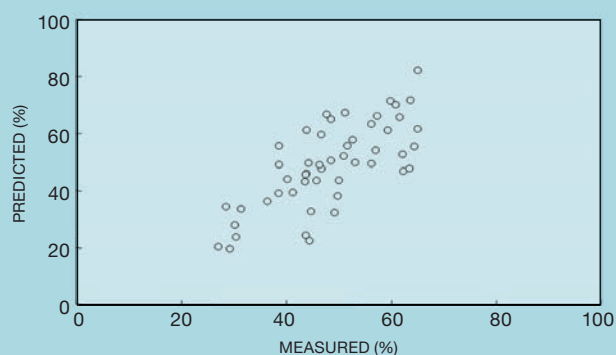


Figure 2: Scatter chart



QUICK READ

- **Accuracy vs. Precision:** Precision (low random error) is measured by R^2 , but accuracy (low bias error) is what truly matters for inferentials
- **Inferential Performance:** The performance index (ϕ) is a better measure than R^2 , reflecting both prediction error and measured property variance. A negative ϕ suggests the inferential is ineffective
- **Re-engineering Triggers:** Monitor inferential accuracy over time; re-engineer if errors consistently exceed the confidence interval or fail to meet accuracy standards

change. Provided it changes in the right direction, it might still be of use, even if only approximately correct. Figure 3 shows that, in this example, the predicted direction of 90% of the changes is correct. And, when it is incorrect (outside the shaded area), it tends to be for the smaller changes.

WE NEED ACCURACY, NOT PRECISION

The main reason why so many poorly engineered inferentials are installed is the misplaced faith that engineers put in R^2 as a good measure of accuracy. It is not. To understand this, we need to separate *precision* from *accuracy*. A *precise* measurement

Figure 3: Reliability of direction of change

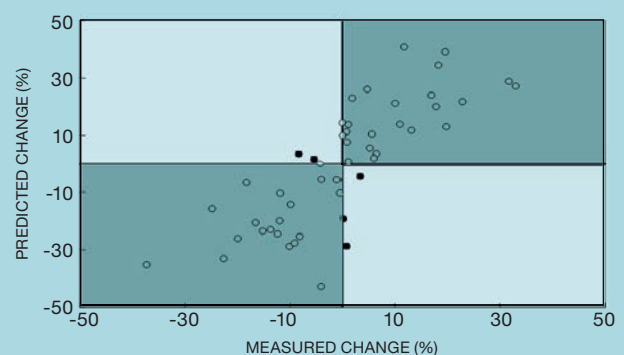
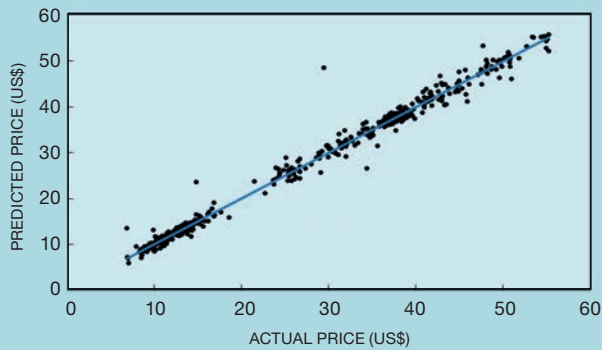


Figure 4: US tech company share price



Figure 5: Inferring tomorrow's share price



has little random error but can have a large bias error. So, a measurement which is consistently wrong by the same amount is precise. An *accurate* measurement has little bias error but can have a large random error. So, a variable measurement which, on average is correct, is accurate. R^2 is a measure of precision and so, if close to 1, tells us that there is little random error but nothing about accuracy.

We can illustrate this with a real-world example. Figure 4 plots the stock price of a well-known US tech company. Those around at the time might remember the company and the issues it had to deal with. Figure 5 shows the results of a stock price predictor developed by yours truly. It's R^2 is 0.989 – apparently very close to perfection. More recently the stock price approached US\$250. So why am I not writing this article on my private island? The answer is illustrated by just one point in Figure 5, where the predicted value was around US\$50, and the actual price was US\$30. It failed to predict the fall that took place in July 1998.

So why is this relevant to predicting product quality? Well, we often install an inferential because the existing quality measurement is a laboratory result that is reported maybe once per day. We want the inferential to give us a much earlier indication of a significant change in quality. If it fails to do so, even

if infrequently, it is of little value. Similarly, an inferential which accurately predicts an unchanging property adds nothing. Whatever parameter we choose to monitor performance must take account of not only how close the inferential matches the measured property, but also how much the measured property changes. One which does this is the *performance index* (ϕ) where:

$$\phi = 1 - \frac{\text{variance of prediction error}}{\text{variance of measured property}} = 1 - \frac{\sigma_{error}^2}{\sigma_{property}^2}$$

To understand how this parameter works, consider first that we have a perfect inferential:

$$\sigma_{error}^2 = 0 \quad \text{so} \quad \phi = 1$$

Now consider an inferential which, on average is correct but never changes (which is clearly of no value). For example, we might predict that the inferential stays at its average value:

$$\hat{y}_i = \bar{y} \quad \text{so} \quad \sigma_{error}^2 = \sigma_{property}^2 \quad \text{and} \quad \phi = 0$$

Now imagine that the measured property is on target and never changes. In this case any error in the inferential causes the controller to wrongly take corrective action – worsening quality control:

$$\sigma_{error}^2 > \sigma_{property}^2 \quad \text{and} \quad \phi < 0$$

So, if we were monitoring ϕ for a working quality controller, we would want to disable the control if ϕ became negative.

Justifying improved control is usually performed by calculating the benefit captured by halving $\sigma_{property}$. If we assume that our control scheme is perfect and the only deviation from target comes from the random error in the quality prediction then, to capture the benefits:

$$\sigma_{error} \leq 0.5\sigma_{property} \quad \text{and so} \quad \phi \geq 0.75$$

Because controllers aren't perfect, we need a significantly higher value for ϕ – typically in excess of 0.85.

COMPARISON WITH R^2

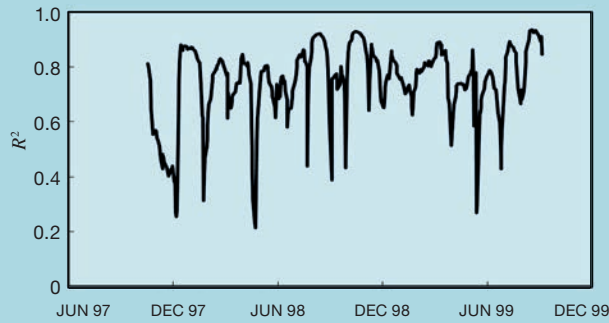
In the last article we showed that, if we changed the coefficients in a single-input linear inferential, R^2 remained unchanged. Consider one of the equations we fitted to the points (2,3), (3,9), and (7,12):

$$\hat{y} = 1.5x + 2$$

This predicts (x,\hat{y}) as (2,5), (3,6.5) and (7,12.5), giving a value of 0.75 for both R^2 and ϕ . If we double the coefficient of x , the prediction becomes (2,8), (3,11), and (7,23). R^2 remains the same, while ϕ becomes significantly negative (-2.57) telling us very clearly to avoid using the inferential.

Similarly, if we were to calculate ϕ for the share price predictor we obtain the value of 0.989 – exactly the same as R^2 . But differences appear if we plot the parameters as rolling values based, for example, on the last 30 records. Figure 6 shows that

Figure 6: Rolling Pearson R²



R^2 shows significant variation but never approaches 0. It tells us that the correlation always exists but not whether it is reliable enough to be used.

Before plotting ϕ we make a minor change to its calculation. In our example, the cause of each prediction error is a change in the actual, rather than predicted, price. So, both σ_{error} and $\sigma_{property}$ change and ϕ therefore changes very little. The solution is to use the previous value of $\sigma_{property}$. So, a better definition is:

$$\phi = 1 - \frac{(\sigma_{error}^2)_n}{(\sigma_{property}^2)_{n-1}}$$

Trending this value, as Figure 7, shows several occasions on which ϕ is negative. If this were an inferential, the composition controller would (on five occasions) take corrective action that would worsen process performance. Despite its almost perfect precision, its lack of accuracy would lead us to reject its design.

MONITORING PERFORMANCE

If we use ϕ to monitor the performance of an installed inferential, we must further modify its calculation. Let us imagine

Figure 7: Rolling performance index

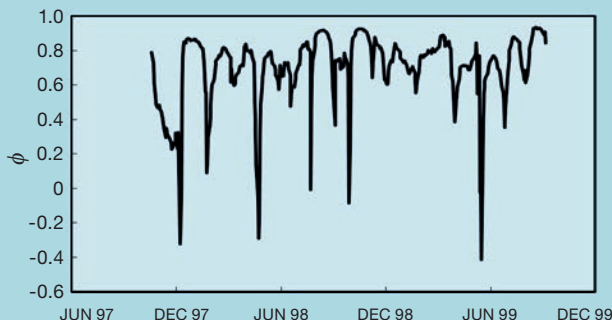
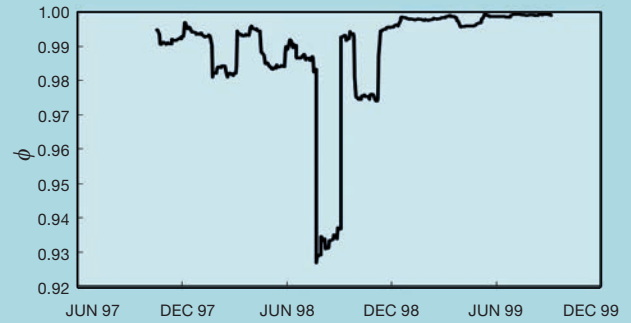


Figure 8: Monitoring inferential performance



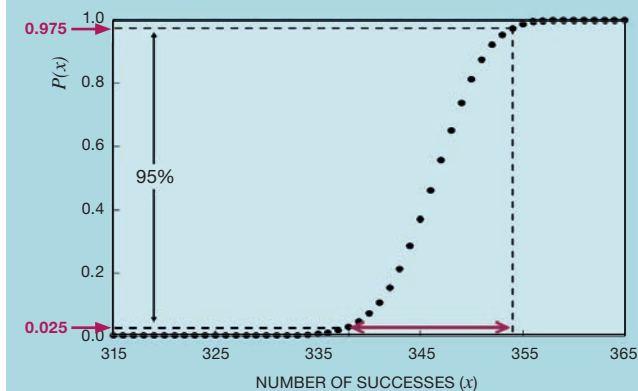
that, at design stage, ϕ was 0.75. In other words, σ_{error} was half of $\sigma_{property}$. Also imagine that, on commissioning, our controller is perfect and achieves the objective of halving $\sigma_{property}$. As a result, ϕ will reduce to 0 – falsely indicating that the inferential has no value. To resolve this, we use a constant value for $\sigma_{property}$, chosen as its value prior to commissioning the control scheme. Figure 8 shows the result, had our share price predictor been in service. It clearly indicates when the failure occurred. However, despite the problem being corrected for the next prediction, ϕ remains low for 30 days. Once a problem is resolved we need to delete the offending record(s) from the rolling calculation.

In addition to detecting inaccuracy as soon as possible, we should monitor performance over a longer period to determine whether the inferential should be re-engineered – maybe because of some change to the process. We monitor this by recording the number of occasions, within a defined time-frame, that the inferential is incorrect. From its development, we know the expected standard deviation of the error (σ_{error}). If the error falls outside the 95% confidence interval (ie $1.96\sigma_{error}$) then we designate this a failure.

The Excel function BINOM.INV(n , p , P) gives the expected number of successes for a defined probability (P) in n



Figure 9: Binomial distribution



independent trials where p is the probability of success in each trial. We might consider a year of daily checks on accuracy and so n is 365. We expect the inferential to be correct 95% of the time, so p is 0.95. We want to know how many of the trials will fall inside the 95% confidence interval, ie inside the range of p between 0.025 and 0.975. See Figure 9.

For example:

$$\text{BINOM.INV}(365, 0.95, 0.025) = 338$$

$$\text{BINOM.INV}(365, 0.95, 0.975) = 354$$

These tell us to expect the inferential to be correct between 338 and 354 days. If the actual number is less than 338 then re-engineering should be considered. If it greater than 354 then the inferential is performing better than expected and perhaps the confidence interval should be reduced to make a more demanding check on accuracy. ■

NEXT ISSUE

In the next issue, we'll cover the pitfalls of automatically updating an inferential based on the latest laboratory result or on-stream analyser measurement.

Myke King CEng FICHEME is director of Whitehouse Consulting, an independent advisor covering all aspects of process control. The topics featured in this series are covered in greater detail in his book Process Control – A Practical Approach, published by Wiley in 2016

Disclaimer: This article is provided for guidance alone. Expert engineering advice should be sought before application.

Abstract deadline: 2 May



Advances

in the Digitalisation of the Process Industries

16–17 October 2025, Manchester, UK

Advances is the must-attend event for industry professionals applying and working with digital technologies and techniques. Join the experts and connect with peers to explore best practice, emergent thinking, and future implications of a dynamic and rapidly changing technology

How to get involved

- Contribute to the conference programme – submit an abstract sharing your automation, process control and digitalisation experience
- Showcase your products and services – talk to us about our range of sponsorship and exhibition packages

Visit the event website to find out more and register your interest in attending.

www.icheme.org/advances

ICHEME