POINTS OF SIGNIFICANCE

Model selection and overfitting

With four parameters I can fit an elephant and with five I can make him wiggle his trunk.

—John von Neumann

In recent months we discussed how to build a predictive regression model[1–3] and how to evaluate it with new data[4]. This month we focus on overfitting, a common pitfall in this process whereby the model not only fits the underlying relationship between variables in the system (which we will call the underlying model) but also fits the noise unique to each observed sample, which may arise for biological or technical reasons.

Model fit can be assessed using the difference between the model's predictions and new data (prediction error—our focus this month) or between the estimated and true parameter values (estimation error). Both errors are influenced by bias, the error introduced by using a predictive model that is incapable of capturing the underlying model, and by variance, the error due to sensitivity to noise in the data. In turn, both bias and variance are affected by model complexity (**Fig. 1a**), which itself is a function of model type, number of inputs and number of parameters. A model that is too simple to capture the underlying model is likely to have high bias and low variance (underfitting). Overly complex models typically have low bias and high variance (overfitting).

Under- and overfitting are common problems in both regression and classification. For example, a straight line underfits a third-order polynomial underlying a model with normally distributed noise (**Fig. 1b**). In contrast, a fifth-order polynomial overfits it—model parameters are now heavily affected by the noise. As we would expect, fitting a third-order polynomial gives the best results, though if the high noise level obscured the actual trend and our goal was to reduce total error, we might choose a less complex model than the underlying model (for example, second-order). The situation is similar for classification—for example, a complex decision boundary may perfectly separate classes in the training set, but because it is greatly influenced by noise, it will frequently misclassify new cases (**Fig. 1c**). In both regression and classification problems, the overfitted model may perform perfectly on training data but is likely to perform very poorly, and counter to expectation, with new data.

To illustrate how to choose a model and avoid under- and overfitting, let us return to last month's diagnostic test to predict a patient's disease status[4]. We will simulate a cohort of 1,000 patients, each with a profile of 100 blood biomarkers and known disease status, with 50% prevalence. Our aim will be to use the cohort data to identify the best model with low predictive error and understand how well it might perform for new patients. We will use multiple logistic regression to fit the biomarker values—the selection of biomarkers to use will be a key consideration—and create a classifier that predicts disease status. For simplicity, we will restrict ourselves to using the F_1 score as the metric; practically, additional metrics should be used to broadly measure performance[4].

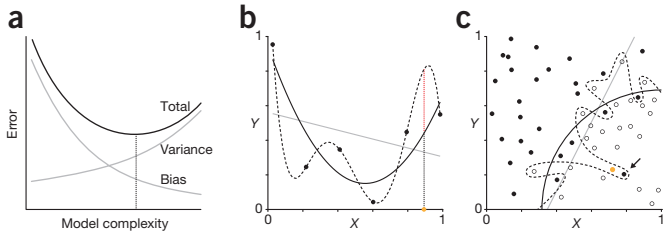

Figure 1 | Overfitting is a challenge for regression and classification problems. (**a**) When model complexity increases, generally bias decreases and variance increases. The choice of model complexity is informed by the goal of minimizing the total error (dotted vertical line). (**b**) Polynomial fits to data simulated from a third-order polynomial underlying a model with normally distributed noise. The fits shown exemplify underfitting (gray diagonal line, linear fit), reasonable fitting (black curve, third-order polynomial) and overfitting (dashed curve, fifth-order polynomial). There is a large difference (red dotted line) in Y prediction at $X = 0.9$ (orange circle) between the reasonable and overfitted models. (**c**) Two-class classification (open and solid circles) with underfitted (gray diagonal line), reasonable (black curve) and overfitted (dashed curve) decision boundaries. The overfit is influenced by an outlier (arrow) and would classify the new point (orange circle) as solid, which would probably be an error.

We might use the data for an entire cohort to fit a model that uses all the biomarkers. When we returned to the cohort data to evaluate the predictions, we would find that they were excellent, with only a small number of false positives and false negatives. The model performed well, but only on the same data used to build it; because it may fit noise as well as systematic effects, this might not be reflective of the model's performance with new patients. To evaluate the model more honestly, we could recruit additional patients, but this would be both time-consuming and expensive. Alternatively, we could split the cohort data into groups: a training set to fit the model, and a testing set (or hold-out set) to estimate its performance. If we applied the common 80/20 split for training and test set, we would randomly select 800 patients for the multiple logistic regression fit and use the remaining 200 for evaluation, with the constraint that both subsets have the same fraction of diseased patients (class balance).

But should we use all the biomarkers for our model? Practically, it is likely that many are unrelated to disease status; therefore, by including them all we would be modeling quantities that are not associated with the disease, and our classifier would suffer from overfitting. Instead, we could construct and evaluate 100 different models, each using 1–100 of the most important biomarkers; we assume that we can estimate this ordering. But even if we do this, we might merely identify the model that best fits the testing set, and thus overestimate any performance metrics on new data.

To gain a more honest assessment of performance, we introduce the validation set. This set is used to evaluate the performance of a model with parameters that were derived from the training set. Only the model with the best performance on the validation set is evaluated using the test set. Importantly, testing is done only once. If the data set is large, the training, validation and test sets can be created with a typical 60/20/20 split, maintaining class balance. When using a small data set, as is common in biological experiments, we can use cross-validation as explained below. In our cohort example, we would train on 600 randomly selected patients, evaluate them using a different set of 200 patients and test only the best model on the remaining 200 patients. After the best model is selected, the final stage of creating a classifier involves

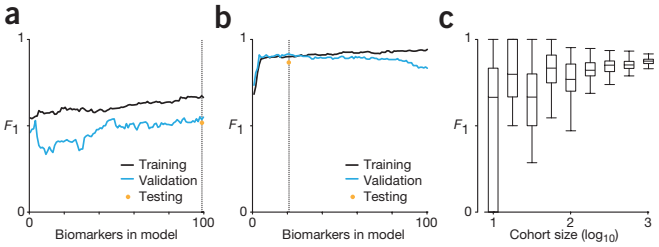

Figure 2 | Select and validate models by splitting data into training, validation and test sets. (**a**) Evaluation of a model built on random data. Performance measured by F_1 scores of 100 models built using multiple logistic regression based on 1–100 random biomarker profiles on a cohort of 1,000 subjects, of whom 50% are affected by the disease. The model with the highest validation F_1 score (dotted vertical line) is evaluated using the testing set (orange dot). The training/validation/test split is 60/20/20. (**b**) Evaluation of a model built with biomarkers. Here we are looking at the same scenario as in **a** but applied to a data set in which the first five biomarkers are correlated with disease. (**c**) The impact of sample size on variation. Tukey-style box plots showing the variation in F_1 score for 100 simulations of cohorts of size 10–1,000 fit using 21 biomarkers selected in **b**. For each round and cohort size, the training/test split is 80/20, with class balance maintained.

recombining the various subsets of data and refitting the model using the selected biomarkers on the entire data set.

To illustrate this entire process and how it can be impacted by overfitting, we simulated our cohort to have random biomarker levels that are independent of disease status and then validated the disease-status prediction of each of the 100 models using the F_1 score[4]. We observed an increase in the training F_1 score as we increased the number of biomarkers (variables in the model) (**Fig. 2a**). This trend is misleading—we were merely fitting to noise and overfitting the training set. This is reflected by the fact that the validation set F_1 score did not increase and stayed close to 0.5, the expected F_1 score for random predictions on this data set.

Suppose now we alter the values of the first five biomarkers to make them predictive by sampling them from a normal distribution with a mean reflecting the disease status ($\mu_{healthy} = 0$, $\mu_{diseased} = 1$ and $\sigma = 1$). Now, the validation F_1 score peaks early with a plateau around 5–25 biomarkers (for our simulation, 21 biomarkers is optimal) and then drops as more biomarkers are included in the model and the effects of overfitting become noticeable (**Fig. 2b**). Because of the possibility of test set overfitting, we expect the best validation F_1 to overestimate the real performance of the classifier. And in this case we do see that the test F_1 (0.85) is below the highest F_1 score (0.91) for the validation set and provides a less biased assessment of performance.

When we randomly selected patients for the training, validation and testing sets, it was with the assumption that each set was representative of the full data set. For small data sets, this might not be true, even if efforts are made to maintain a balance between healthy and diseased classes. The F_1 score can vary dramatically if 'unlucky' subsamples are chosen for the various sets (**Fig. 2c**). If increasing the sample size to mitigate this issue is not practical, *K*-fold cross-validation can be used.

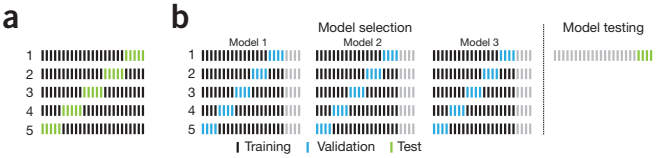

Figure 3 | *K*-fold cross-validation involves splitting the data set into *K* subsets and doing multiple iterations of training and evaluation. The metric (for example, F_1 score) from all iterations is averaged. (**a**) A strategy with $K = 5$ without model selection. Training sets and test sets are used to derive prediction statistics. (**b**) Nested *K*-fold cross-validation with model selection. This strategy uses a validation set for model selection using the strategy of **a**. The best model is then tested on the separate test set. Gray bars indicate samples not used at the represented stage.

K-fold cross-validation leverages information in small data sets by combining the results of *K* rounds of different training, validation and test splits of the same data set. For example, for $K = 5$ (*K* is commonly set to 5 or 10), we would create five balanced 80/20 training and test splits (**Fig. 3a**) and average the test F_1 scores from each round. This scheme applies if a single model is to be tested.

For multiple models, we apply the scheme analogously to the training and validation scenario (**Fig. 3b**). First, we reserve a portion of the data set for testing the best model, which we find by calculating the average F_1 score for each model from *K* rounds of training/validation splits. Because the performance metric, such as the F_1 score, is calculated several times, *K*-fold cross-validation provides an estimate of its variance.

When *K* is set to the number of samples *N*, the approach becomes leave-one-out cross-validation (LOOCV). This can be attractive, as it further reduces the likelihood that a split will result in sets that are not representative of the full data set. Furthermore, there are tricks to make this computationally feasible, given the large number of models that would need to be fit. However, this approach is known to overfit for some model-selection problems, such as our problem of selecting the number of biomarkers[5].

Finding a model with the appropriate complexity for a data set requires finding a balance between bias and variance. It is important to evaluate a model on data that were not used to train it or select it. Small sample sizes, common to biological research, make model selection more challenging. Here we have shown that test set and cross-validation approaches can help avoid overfitting and produce a model that will perform well on new data.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

Jake Lever, Martin Krzywinski & Naomi Altman

1. Altman, N. & Krzywinski, M. *Nat. Methods* **12**, 999–1000 (2015).
2. Krzywinski, M. & Altman, N. *Nat. Methods* **12**, 1103–1104 (2015).
3. Lever, J., Krzywinski, M. & Altman, N. *Nat. Methods* **13**, 541–542 (2016).
4. Lever, J., Krzywinski, M. & Altman, N. *Nat. Methods* **13**, 603–604 (2016).
5. Shao, J. *J. Am. Stat. Assoc.* **88**, 486–494 (1993).

Jake Lever is a PhD candidate at Canada's Michael Smith Genome Sciences Centre. Martin Krzywinski is a staff scientist at Canada's Michael Smith Genome Sciences Centre. Naomi Altman is a Professor of Statistics at The Pennsylvania State University.